جامعة هواري بومدين
للعلوم والتكنولوجيا
U S T H B

Université des Sciences et de la Technologie Houari Boumediene

Faculté des Mathématiques

Département de Recherche Opérationnelle

Laboratoire : LaROMaD

Thèse de Doctorat LMD en Mathématiques Appliquées, Spécialité : Recherche Opérationnelle et Management

Présentée par[1] :

DEY Imene

Thème

Méthodes coopératives à base de population pour le problème des réseaux de transport et de télécommunication

Soutenue publiquement le 24/02/2024, devant le jury composé de :

Mme. Isma Dahmani	MCA, USTHB, Alger,	Présidente
M. Méziane Aider	Professeur, USTHB, Alger	Directeur de thèse
M. Mhand HIFI	Professeur, UPJV, Amiens, France	co-Directeur de thèse
M. Abdelkader Merakeb	Professeur, UMMTO	Examinateur
M. Rachid Ouafi	Professeur, USTHB, Alger	Examinateur
M. Mohamed Aidene	Professeur, UMMTO	Invité

Année Universitaire : 2023/2024

University of Science and Technology Houari Boumediene

Faculty of Mathematics

Department of Operational Research

Laboratory: LaROMaD

LMD Doctoral Thesis in Applied Mathematics,

Specialty: Operational Research and Management

Presented by[1]:

DEY Imene

Theme

Population-based cooperative methods for the problem of Transport and telecommunications networks

Publicly graduated in 24th February 2024, in front of the jury composed of:

Mrs. Isma Dahmani	MCA, USTHB, Alger,	President
Mr. Meziane AIDER	Professeur, USTHB, Alger	Thesis Director
Mr. Mhand HIFI	Professeur, UPJV, Amiens, France	Thesis Co-Director
Mr. Abdelkader Merakeb	Professeur, UMMTO	Examiner
Mr. Rachid Ouafi	Professeur, USTHB, Alger	Examiner
Mr. Mohamed Aidene	Professeur, UMMTO	Invité

College Year: 2023/2024

Dedications

This research is lovingly dedicated to my beloved mother **Leila Karadaniz**, my beloved father **Redouane Dey,** my precious sisters **Manel & Sarah** my awesome brother **Aymen**,

and my beloved husband **Ilyes Aiouche**,

They have always been a source of inspiration.

They give me all the encouragement and guide to achieve my goal. Without their love and support this would not have been made.

Acknowledgement

First of all, I want to thank Allah, The Total Provider, The All-Knowing One, for guiding me to knowledge and giving me the strength and the health to achieve my goals, ALhamdulilah.

My deepest thanks to my supervisor,Mr **Mhand Hifi**, professor at UPJV, France, for their unwavering support and guidance throughout my PhD's study. His expertise and patience have been invaluable to me and have played a crucial role in the success of this thesis. Words are not enough to express how much I'm thankful for all the efforts and help he provided. May Allah protect him and bless him. I'm looking forward to work with him again.

I am grateful to our research Laboratory **LAROMAD** for providing me with the opportunity to conduct my research and for all of the resources and support they provided. I would like to extend a special thanks to my supervisor, Mr **Aider Meziane**, a professor at USTHB, who went above and beyond to help me with my work. I would like to thank him for all the guide and help he gives me and for being always present.

I address my thanks to my professor, Mr. Mustapha Chelali, Professor at University of Blida, for sharing his knowledge with me, as well as, for his many pieces of advice and valuable ideas.

A limitless thanks for my great husband, without his support and encouragement I would never achieve this high degree.

I am deeply thankful to awesome family, Mom, Dad, my two sisters and my brother, for their love and support during this process. Without their encouragement and motivation, I would not have been able to complete this journey.

I would also like to thank my colleagues at USTHB for their support and collaboration during my research.

Abstract

In this thesis, a new NP-hard combinatorial optimization variant of the maximal covering location problem is considered, called the Fuzzy Capacitated Maximal Covering Location Problem, where three different hybrid metaheuristics have been tackled to approximately solve it. An instance of the problem is represented by a set of customers in a given network with their distances, such that the coverage degree of facilities and the distance between customers are fuzzy. Its goal is to open or select a subset of facilities to be located near customers so that their coverage is maximized. The first approach is an evolutionary method based upon the grey wolf optimizer (GWO), which starts by generating an initial population using a greedy rule strategy that is able to achieve feasible solutions according to the current positions of wolves. In the second method, we propose a modified gray wolf optimizer combined with a series of local searches and an upper bound delimiting wolf movements. The method enhances the quality of the solutions belonging to the current population by adding a series of local searches. Further, the exploration strategy is based on the drop-and-rebuild operator for jumping from the current region to un-visited ones. And in the last resolution approach, the algorithm starts by applying the basic rounding procedure, which establishes an initial feasible solution for the problem. Subsequently, an interval search is conducted, incorporating both lower and upper limits, to constrain the search process and improve the quality of the solution. The algorithm follows an iterative search approach, resembling a descent method, to refine the solution iteratively. All the proposed resolution methods have been tested on a set of benchmark instances commonly used in the literature. The results were compared to those achieved by recent methods from the literature as well as the state-of-the-art Cplex solve. Promising and encouraging results have been provided, for which we have experimentally proved the effectiveness of our aforementioned resolution techniques.

Keywords: Metaheuristics, Combinatorial optimization, Hybrid , Evolutionary

Résumé

Dans cette thèse, nous nous intéressons à l'étude d'une nouvelle variante d'un problème NP-difficile d'optimisation combinatoire, qui est le problème de localisation de couverture maximale, notée MCLP. Cette variante de MCLP est appelée problème de localisation de couverture maximale à capacité floue, où trois différentes métaheuristiques hybrides ont été abordées pour le résoudre approximativement. Une instance du problème est représentée par un ensemble de clients dans un réseau donné avec leurs distances et tel que le degré de couverture des stations et la distance entre les clients sont flous. Son objectif est d'ouvrir (sélectionner) un sous-ensemble de stations à localiser de manière à maximiser la couverture totale des clients. La première approche est une méthode évolutive basée sur l'optimiseur de loup gris (GWO), qui commence par générer une population initiale à l'aide d'une stratégie gloutonnes capable d'obtenir des solutions réalisables en fonction de la position actuelle des loups. Dans la deuxième méthode, nous proposons un optimiseur de loup gris modifié, combiné à une série de recherches locales et une borne supérieure délimitant les mouvements des loups. Cette méthode améliore la qualité des solutions appartenant à la population actuelle en ajoutant une série de recherches locales. De plus, une stratégie d'exploration a été introduite à notre approche, basée sur un opérateur de suppression/ reconstruction pour diversifier l'espace de recherche et visiter de nouveaux sous-espaces. Dans la dernière approche de résolution, l'algorithme commence par appliquer la procédure de rounding, qui établit une solution initiale réalisable pour le problème. Ensuite, une recherche par intervalles est effectuée, intégrant à la fois des bornes inférieures et supérieures, afin de limiter le processus de recherche et améliorer la qualité de la solution. L'algorithme suit une approche de recherche itérative , ressemblant à une méthode de descente, pour affiner la solution d'une manière itérative. Les résultats trouvés ont été comparés à ceux obtenus par les méthodes récentes de la littérature ainsi que le solveur CPLEX. Des résultats encourageants ont été fournis, montrant l'efficacité de nos techniques de résolution pour le problème de localisation de couverture maximale à capacité floue.

الملخص:

في هذه الأطروحة، تم النظر في متغير أمثلية اندماجي جديد من نوع NP-hard لمشكلة موقع التغطية القصوى ، يسمى مشكلة موقع التغطية القصوى المضبب(Fuzzy Capacitated Maximal Covering Location Problem)، حيث تم معالجة أربعة أساليب ميتاهيرستية هجينة مختلفة لحلها تقريبًا. تم تمثيل مثال للمشكلة بواسطة مجموعة من العملاء في شبكة معينة مع مسافاتهم بحيث تكون درجة تغطية المرافق والمسافة بين العملاء غير واضحة. هدفها هو فتح (اختيار) مجموعة فرعية من المرافق التي سيتم تحديد موقعها على العملاء بحيث يتم تعظيم تغطية العملاء إلى الحد الأقصى. النهج الأول هو أسلوب تطوري يعتمد على مُحسِّن الذئب الرمادي (GWO)، والذي يبدأ بتوليد مجموعة سكانية أولية باستخدام استراتيجية القاعدة الجشعة القادرة على تحقيق حلول مجدية وفقًا للمواقف الحالية للذئاب. في الطريقة الثانية، نقترح مُحسِّن الذئب الرمادي المعدل مقترنًا بسلسلة من عمليات البحث المحلية والحد الأعلى لحركات الذئب. تعمل الطريقة على تحسين جودة الحلول التي تخص السكان الحاليين عن طريق إضافة سلسلة من عمليات البحث المحلية. علاوة على ذلك، تعتمد استراتيجية الاستكشاف على عامل الإسقاط/إعادة البناء للانتقال من المنطقة الحالية إلى المناطق التي لم تتم زيارتها. في التقنية الثالثة، تم تضمين إجراء التقريب الأساسي الذي يوفر مجموعة أولية أفضل من (GWO). وفي نهج الحل الأخير، تبدأ الخوارزمية بتطبيق إجراء التقريب الأساسي، الذي يضع حلاً أوليًا ممكنًا للمشكلة. وبعد ذلك، يتم إجراء بحث على فترات، يتضمن الحدود الدنيا والعليا، لتقييد عملية البحث وتحسين جودة الحل. تتبع الخوارزمية أسلوب بحث تكراري، يشبه أسلوب النزول، لتحسين الحل بشكل متكرر. تم اختبار جميع طرق الحل المقترحة على مجموعة من الأمثلة القياسية المستخدمة بشكل شائع في الأدبيات. وتمت مقارنة النتائج التي تم العثور عليها مع تلك التي حققتها الأساليب الحديثة للأدبيات بالإضافة إلى حل (Cplex) الحديث. وقد تم تقديم نتائج واعدة ومشجعة، والتي أثبتت تجريبيا فعالية تقنيات الحل المذكورة أعلاه.

Contents

List of Figures

List of Tables

Introduction

Contexte

As scientists, engineers, and managers, we always have to take decisions. Decision-making is everywhere. As the world becomes more complex and competitive, making a decision must be tackled in a rational and optimal way (Talbi (2009)).

The integer programming (IP) problems occupy large interests in operational research and computer science fields. It has uncountable applications in real-life phenomena, such as the Traveling Salesman problem, Scheduling, Graph's coloration, Knapsack problem, etc.
One of the most famous IP problems is the 0-1 integer programming problem, where all its variables are binary.

TThe branch of facility location problem (FLP), also known as location analysis, is one of the most famous decision problems in operations research and precisely the IP problem, which consists of finding the optimal placement of facilities to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing and competitors' facilities. They also exist in a maximization problem version, for which the interest is to maximize the total coverage provided by the installed facilities.

The maximal covering location problem, namely MCLP, appears in various life domains. We can cite its applications in seeking the best location of emergency response facilities, services, and vehicles (Li et al., 2011), communication networks (Lee and Murray, 2010), security monitoring sensors (Murray et al., 2007) and, retail stores (Frank and Lieselot, 2007). The maximal covering location problem consists of determining the optimal subset of facilities to be installed in a given location network, aiming to cover the maximal number of customers,such that each facility's coverage area and the distance between a customer and a facility are fixed. The NP-hard combinatorial optimization problem MCLP is also available in reality with several variants; as an example, one can observe the constraints considered for the mobile phone network and the web-internet networks (Davari et al., 2011),and location of Wi-Fi antennas (Lee and Murray, 2010).
The MCLP is based on finding the best facility locations to cover the maximum number of nodes (population) and ignoring the fact that, in a major realistic problem, each facility has a finite capacity of coverage or a limited distance of coverage, such as emergency services such as police, ambulances, hospitals, etc. However, in the literature, there exist many variants of the maximal covering location problem that, indeed, consider the finite capacity of a facility, such as The Capacitated Maximal Covering Location Problem (CMCLP), Pirkul and Schilling (1991).
Thus, some other researchers concluded that it is likely important to include the fuzzy distance between a facility location and the customers as a new interesting constraint. In fact, a real-life example can be seen in illumination sources. It has a radius where the light can totally be distributed for clients within it, and the light decreases whenever the distance between the light source and the client increases.

In this thesis, we are interested in studying an extended version of the MCLP with fuzzy distances, known as the Fuzzy Capacitated Maximal Covering Location Problem (namely FCMCLP) (Atta et al., 2022). The main goal of the FCMCLP is to select the best subset of facility locations that covers the maximum number of customers in a given network, such that both the distance between the customers and the installed facilities and the coverage area of each facility are fuzzy. Because of the complexity of the MCLP, the FCMCLP, defined in medium-scale and large-scale networks, cannot be solved in polynomial time, and it's classified as an NP-hard problem.

For that purpose, we proposed hybrid algorithms based on the population's heuristics to approximately solve the fuzzy capacitated maximal covering location problem.

Contribution

This thesis has been built on the following major contributions:

In the first proposed method, we investigate the use of an evolutionary algorithm for tackling the fuzzy capacitated maximal covering location problem. The proposed evolutionary method is based upon the grey wolf optimizer, which starts by generating an initial population using a greedy rule strategy that is able to achieve feasible solutions according to the current positions of wolves. In order to enhance the quality of solutions induced, a series of local searches are added to explore the search space by exploiting some nice strategies. The behavior of the method is computationally analyzed on a set of instances from the literature. Encouraging results have been provided.

Aiming to achieve better results and develop a new fast technique, a hybrid population-based algorithm is proposed to solve the fuzzy capacitated maximal covering location problem (FCMCLP). An instance of the problem is represented by a set of customers in a given network with their distances, such that the coverage degree of facilities and the distance between customers are fuzzy. Its goal is to open or select a subset of facilities to be located near customers so that their coverage is maximized. Two versions of a hybrid population-based algorithm are designed for tackling the aforementioned problem: (i) a modified grey wolf optimizer combined with exploration and exploitation strategies, and (ii) a modified grey wolf optimizer combined with a series of local searches and an upper bound delimiting wolf movements.
Each version of the method enhances the quality of the solutions belonging to the current population by adding a series of local searches. Further, the exploration strategy is based on the drop-and-rebuild operator for jumping from the current region to un-visited ones. Finally, the behavior of the proposed method is evaluated on all the benchmark instances of the literature, where its provided results are compared to those reached by more recent methods of the literature and the state-of-the-art Cplex solver. The hybrid population-based algorithm came up with several new bounds.

On the other hand, a new mathematical approach was proposed to mainly beat all the existing metaheuristic bounds. In order to achieve the aforementioned goal and approximately solve the FCMCLP problem, an extended rounding strategy-based algorithm is designed. The algorithm starts by applying the basic rounding procedure, which establishes an initial feasible solution for the problem. Subsequently, an interval search is conducted, incorporating both lower and upper limits, to constrain the search process and improve the quality of the solution. The algorithm follows an iterative search approach, resembling a descent method, to refine the solution iteratively.
Finally, in order to evaluate the effectiveness of the proposed method, its provided results are compared to those achieved by recent methods in the literature as well as the state-of-the-art Cplex solver. These experiments are conducted on a set of benchmark instances commonly used in the literature. The comparative analysis demonstrates promising and encouraging results obtained by the proposed method.

Organization

Our work is organized as follows: In the first chapter, we have suggested some definitions and explanations that need to be clarified in order to well understand the problem and the proposed method. First, we started by defining the combinatorial optimization domain, then we explained its algorithmic complexity. Moreover, we show a famous real-life example: the traveling salesman problem. In the last section, we tackle different resolution techniques proposed by the literature, starting with the Branch and Bound method, then we pass through several single solution-based metaheuristics (local search, tabu search) and population-based metaheuristics: particle swarm optimization. Finally, we give a brief explication of the two recent resolution approaches: hybrid metaheuristics and matheuristics.
In the second chapter, deep research into the related works has been established. We started by presenting the Maximal Covering Location Problem and illustrated its mathematical formulation. Then we've passed through different resolution techniques used by the literature researchers. A large-scale MCLP has also been taken into consideration. In the next section, we jumped to the different variants of the MCLP, beginning with the Capacitated Maximal Covering Location problem (CMCLP) and its related works. After that, we modeled the fuzzy maximal covering location problem (FMCLP) and showed a real-life illustrative example. At last, the mathematical formulation of the Fuzzy Capacitated Maximal Covering Location Problem (FCMCMLP) has been given with details, with the aim of illustrating the thesis' studied problem.

The next three chapters , chapter 3, 4 , and 5, represents our contributions and works :

In the third chapter 3, we started by defining the fuzzy capacitated maximal covering location problem (FCMCLP) and gave its mathematical description. A basic definition of the grey wolf optimizer (GWO) was included. The work is based on a hybrid algorithm for FCMCLP, for which two versions were constructed. They generally consist of creating a starting population with or without an upper bound. Then, we show a brief explicating algorithm that aims to generate new wolf positions. Hereafter, both intensification and avoiding cycling strategies were defined, consisting of using the 2-opt operator and Tabu search. A drop-and-rebuild technique was introduced before the computational results. We have tested the behavior of both versions and the effect of the upper bound on the quality of the population. Also, population size has been an important parameter in the study.

The forth chapter 4, presents a brief formal description of the studied problem. Then, an evolutionary algorithm based upon the Grey Wolf Optimizer It begins by defining the aforementioned metaheuristic; from there on, we give a greedy algorithm that contains the principle steps of generating a GWO initial population. A simple 2-opt intensification strategy and the Drop and Rebuild approach have been included, aiming to achieve un-visited solutions in the search space. Even though the chapter was an independent study of the Grey Wolf Optimizer approach on the FCMCLP, encouraging results have been found and resumed in the computing test section.

In chapter 5, a formal description of FCMCLP is presented. Next, a basic rounding procedure has been introduced and explained. Then, the interval search technique constructed by solving two mixed-integer programming problems is presented in order to provide both lower and upper bounds and construct a new valid constraint for the FCMCLP model. Additionally, an enhanced version of the algorithm is presented in the same section. The next section contains a strategy that aims to bound the space with cardinality constraints based on solving two subproblems, and then a global algorithm has been added to resume the whole proposed solving method. Finally, in the Results section, we evaluate the performance of the designed rounding procedure by testing and analyzing its behavior on instances sourced from existing literature. The achieved results are then compared to those provided by recent methods from the literature and the state-of-the-art Cplex solver.

Chapter 1

Generalities

1.1 Introduction

As scientists, engineers, and managers, we always have to take decisions. Decision-making is everywhere. As the world becomes more complex and competitive, making a decision must be tackled in a rational and optimal way. Talbi (2009)
In this chapter, we'll be talking about combinatorial optimization (namely CO), which is a large field of decision-making problems. First, we define combinatorial optimization (CO) and show some real-life examples. Then, a detailed part of CO resolution approaches is introduced to explain more about exact and approximate solving methods. Next, we tackle some heuristic and meta-heuristic techniques. Finally, we give a particular explanation about the different 'Matheuristics'

1.2 Combinatorial Optimisation

1.2.1 Definition

Combinatorial optimization, also known as a branch of integer programming, is the process of taking a decision that optimally solves a given problem. Mathematically defined as the process of seeking the maximum (or minimum) of an objective function, which is defined in a discrete large domain, Papadimitriou and Steiglitz (1998).
An Integer Linear Programming can be formulated as follows: (We consider the maximisation case).

$$\max \quad c^T x \tag{1.1}$$

$$\text{s.t.} \quad Ax = b \tag{1.2}$$

$$x \geq 0 \tag{1.3}$$

$$x \qquad integer \tag{1.4}$$

Such that:
c : is the objective function vetor.
The set 1.2 refers to the constraints. And the constraints 1.3 insures the non-negativity of the

integer decision variable x.
We can also note : $D = \{x \text{ integer}/Ax = b \text{ and } x \geq 0\}$, called the domain of study.

1.2.2 Integer Linear Programming Example

There are many real-life examples that incarnate the combinatorial optimization problems. In this part, we cite one of the well-known integer linear programming problems:

1.2.2.1 Travelling Salesman Problem (TSP)

The traveling salesman problem (or also vehicle routing problem), namely TSP, is an integer linear programming problem that consists of finding the shortest path from an origin city to visit $N-1$ given cities exactly one time and then returning to the origin city, Cook et al. (2011).

It can be resumed in the following question : **Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city**

It was first mathematically formulated in the 1930s by Flood (1956), aiming to solve a school bus routing problem.It was also known by **Hassler Whitney** as the "48 states problem". The earliest publication was named by "travelling salesman problem" was the 1949 RAND Corporation report by **Julia Robinson**,Lawler et al. (1986).

TSP can be represented by an undirected graph, in which the N cities are modelled by the nodes. The path between each two nodes represents the distance between their constituted cities.
We note c_{ij} the distance (or cost) between city i to city j.
Let x_{ij} be the decision variables, such that $i, j \in V = \{1, ..., N\}$.

$$x_{ij} = \begin{cases} 1, & \text{If the path passes from city } i \text{ to city } j \\ 0, & \text{Otherwise} \end{cases}$$

The mathematical modeling of the the TSP is given as follows:

The objective function 1.5 minimizes the total path distance (or cost). Constraints 1.6 and 1.7 makes sure that for each city i can only be visited (enter and exit city i) once.

$$\min \quad z(x) = \sum_{i \in V} \sum_{j \in V, j \neq i} c_{ij} \times x_{ij} \tag{1.5}$$

$$\text{s.t.} \quad \sum_{i \in V, i \neq j} x_{ij} = 1, \forall j = 1, ..., N \tag{1.6}$$

$$\sum_{j \in V, j \neq i} x_{ij} = 1, \forall i = 1, ..., N \tag{1.7}$$

$$x_{ij} \in \{0, 1\}, \forall i, j \in \{1, ..., N\} \tag{1.8}$$

Herein, in figure 1.1 , a complete 4 nodes graph is illustrating the travelling salesman problem:

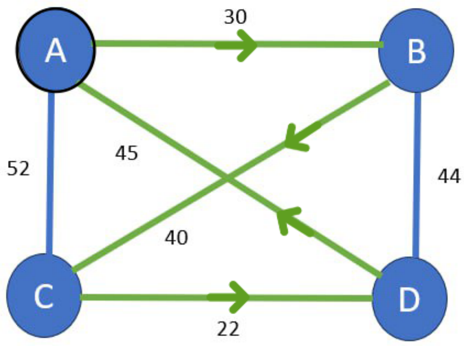

Figure 1.1: An optimal solution for the Travelling Salesman Problem

An optimal path by considering the origin city **A**, is colored in green, which is : {A,B,C,D,A}, with minimal objective value : **137**

1.3 Combinatorial optimization resolution techniques

At a first sight, the obvious resolution technique of an integer programming problem is the enumeration of its possible solutions. Which can directly lead to an optimal best solution.
Unfortunately, the enumeration cannot be used for the combinatorial optimization problems with large solutions sets. For example, in figure 1.1, we can see that we handled the TSP with a graph that contains only four nodes, which made the enumeration much simpler.
If we consider a complete graph with **n** nodes, than the number of possible tours is computed as $\frac{(n-1)!}{2}$.
One can easily observe, that this enumeration is very exhaustive whenever the problem get bigger. It's only restrictive to the small cases.

For that purpose, in the next part, we introduce some exact and approximate resolution techniques for the combinatorial problems.

1.3.1 The Branch and Bound (B&B)

The Branch and Bound (namely BB, or B&B), is a well-known method used to solve the optimisation problems, by splitting the original problem into smaller sub-problems, easier either to solve or to eliminate as an unsolvable sub-problem Papadimitriou and Steiglitz (1998); Levitin (2005). It's generally used to treat the discrete and combinatoirial optimization problems.
The method was first proposed by Ailsa Land and Alison Doig in 1960 for discrete programming, Land and Doig (2010). And it was first applied on the Travelling Salesman Problem (TSP) . See section 1.2.2.1. (Readers can refer to Little et al. (1963))

The main idea of the B&B method is, instead of enumerating all feasible solutions, we separate the candidate solutions using a rooted tree, by defining a separate procedure. The algorithm explores the branches of this tree, which represent subsets of the solution set.
Then, we evaluate the upper and lower bound corespondent to the candidate nodes of the tree.

The last step in B&B is to eliminate the non-feasible nodes or the nodes that cannot lead to a better solution (sterile node). And then, to determine the nodes that provide optimal solutions.

1.3.2 Approximate Algorithms

In the majority of the operational research and computer science fields , approximation approaches take an important part in solving difficult problems. They consist of preparing efficient algorithms that aim to provide approximate solutions to optimization problems. Readers are referred to Williamson and Shmoys (2011)
The approximate methods ,are divided in two sub-classes: **Heuristics and Metaheuristics** .

The **heuristics** find "good" solutions for large size problem instances.But, generally, they cannot insure the approximation on the found bounds. The **Metaheuristics**, can be applied to solve any optimization problem, by giving a guide strategy using heuristics to solve specific optimization problem.

In other words, **Heuristic** is a technique that uses the studied problem information to find "good" solution . Meanwhile, **Metaheuristics**, aim to find promising solutions for a given problem, but without specifying the problem, and they provide an algorithm that can deal with different problems.
For more details, readers are refereed to see Talbi (2009).

In the next part, we give some important heuristics and metaheuristics techniques that are related to the combinatorial optimization, and specially with integer linear programming.

1.3.3 Single-solution based Metaheuristics

Herein, we describe some of the famous single-solution based metaheuristics. The single-solution approach, starts with a given feasible solution for the problem at hand, and then, it aims to improve the quality of this solution. Such as : **Local Search, Variable Neighborhood Search, Tabu Search, Simulated Annealing, etc.**

1.3.3.1 Local Search

It's the oldest and the simplest well-known metaheuristic. It's mainly used to solve hard computational optimization problems.
The principal of the local search, consists of using an initial solution, and at each iteration, the heuristic searches within its neighborhood, for a new solution , that has a better objective value. The LS algorithm, keeps jumping from a feasible solution to another, in the current search space, and stops, when no new improved solution is found. The last best solution is called "optimal" for the given problem .

The hard question to pose, while using the local search method is how to select the best neighbor for a given solution? There are many strategies, that can help to find the better neighbor: starting with the **best improvement (steepest descent)** which consists of selecting the neighbor with the best objective, **first improvement** selects the first improving neighbor found, and **random selection** choose a random solution from the neighborhood.

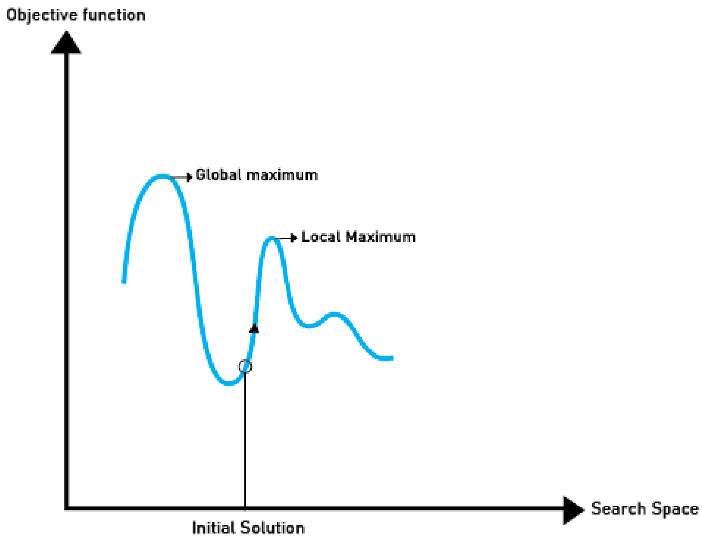

Figure 1.2: Local Search Steepest descent

The major disadvantage of the local search is that it can easily stop at a local optimum. See figure **??**.For that purpose, many improving heuristics have been generated in order to avoid this serious optimization problem.

1.3.3.2 Tabu Search

The Tabu Search is a popular metaheuristic, it was first created by **Glover, in 1986**. It behaves like the local search algorithm with the steepest improvement strategy, but, at each step, even worse neighbors can be taken, in order to escape from a local optima caused by non improving solutions.
Diversification: Tabu Search use prohibitions that aim to forbid the search by selecting previous-visited solutions. It organises those forbidden (namely, Tabu) solutions, in a list, called the **Tabu List**. It's also possible to set up rules that the user provided in order to add even more "Tabu" solutions on it.
Furthermore, within a previous described user condition, called **Aspiration Criterion**, tabu solutions can be admissible, such as: selecting a tabu solution to be accepted, if it has a better objective value among the current set of solutions.

A full descriptive Tabu search algorithm is resumed bellow 1 .

Input. A starting solution X.
Output. X^* : the best solution in the search space.
1: Initialize the tabu list, the aspiration criterion.
2: **repeat**
3: Select the best (non tabu or asperation criterion admissible) neighbor $\overline{X}$.
4: Update $X = \overline{X}$.
5: Apply both intensification and diversification processes.
6: **until** Stopping criteria met.

Algorithm 1: - A Tabu Search Algorithm (Talbi (2009))

For more details about the Tabu search, readers can view Laguna et al. (1991).

1.3.4 Population-based Metaheuristics

In the following section, a brief definition about the Population-based Metaheuristics is given to explain one of the most popular metaheuristics : Particle Swarm Optimization, (namely PSO).

The Population-based Metaheuristcis, can be viewed as a series of improvements based on an initial population of solutions. Then, a new population is generated and integrated into the current one, pursuing some selection and evaluation procedures, which aim to replace the current population with a new population that contains the elite solutions found so far.

1.3.4.1 Particle Swarm Optimization

The Particle Swarm Optimization is a stochastic population-based metaheuristic generally applied on continuous optimization problems. It's inspired from natural organisms such as bird flocking, fish schooling, etc. Clerc (2010).
With a basic view of PSO, it is defined by a swarm (which represents the population) of N particles (feasible solutions of the given problem) on a search space.
Each particle is represented by:
A vector x_i of the current position of the particle in the search space.
A position vector p_i that represents the best solution found by the particle.
A velocity vector v_i that contains the direction for which the particle will travel in.
Two fitness values: first one is the fitness of x_i , the second fitness is for p_i vector.

At each iteration t, each particle adjusts its position $x_i(t)$, toward the global optimmum using the following equations:
Let p_g be the best position found by the whole swarm.

$$x_i(t) = x_i(t-1) + v_i(t) \tag{1.9}$$

Such that:

$$v_i(t) = w \times v_i(t-1) + \rho_1 \times (p_i - x_i(t-1)) + \rho_2 \times (p_g - x_i(t-1)) \tag{1.10}$$

Where ρ_1 and ρ_2 are two randoms in $[0, 1]$. And w is the inertia weight.
Then the selection of the best found particle is tested according to the global best fitness found in the swarm.
For furthermore details, interested readers can refer to Talbi (2009).

1.3.5 Hybrid Metaheuritics

For the purpose of finding an "optimal" solutions (best solutions) to NP-hard optimization problems, researchers developed a new sort of metaheuristics, called the **Hybrid Metaheuritics**. They can be defined as a combination of algorithms between metaheuristics and machine learning techniques. It was highlighted that the best found results for both classical and real-life optimization problems are provided by these powerful search algorithms.Talbi (2009); Voss et al. (2009); Gendreau et al. (2010).

It has already been mentioned, that simple metaheuristics cannot guarantee the best performance for every problem. So, the hybrid metaheuristics have been implemented to exploit different techniques for combining many heuristic algorithms.

1.3.6 Matheuristics

It has already been shown that hybrid metaheuristics are a combination of different heuristic algorithms and machine learning techniques. In this section, a brand new sort of combination is described, called the **Matheuristics**.
Matheuristics are a combination between metaheuristics and exact algorithms. It can also be viewed as the use of mathematical programming techniques on different optimization problems to obtain heuristic solutions.
Recently, it has been an attractive topic for different researchers. Many mathematical programming approaches have been implemented in the matheuristics, such as: Relaxation, decomposition methods, Bender's decomposition, cutting and pricing algorithms, etc.
(See Voss et al. (2009); Lagos et al. (2016), Maniezzo et al. (2021)).

1.4 Conclusion

In this chapter, we have defined and summarized some important operational research generalities, that helps understand the following chapters and works. First, we begin by defining the combinatorial optimization and show a real-life problem example. Then, a brief resolution technique study has been introduced, showing different exact and approximate approaches. We started by explaining the Branch and Bound method, then we explained the difference between heuristics and metaheuristics. After that, we gave a brief view about the most popular metaheuristics: Local search, Tabu Search, Particle Swar Optimization. Finally, a small explanative part has been added to describe the main goal of using and implementing both hybrid metaheuristics and Matheuristics.

Chapter 2

The Maximal Covering Location Problem and its variants

2.1 Introduction

Within the different fields of operational research, the facility location problem represents an important combinatorial optimisation problem, that consists in determining optimal positions of facilities in order to either minimize the transportation costs, or to maximize the coverage of customers, which is also known as the maximal covering location problem (namely MCLP).

In this chapter, we are interested in defining the maximal covering location problem with some of its variants. We also show some interesting works from the literature that have been proposed to solve the MCLP.

2.2 The maximal covering location problem

The maximal covering location problem (MCLP), is an NP-hard integer linear problem, aims to identify the maximum population of demand nodes, which can be served within a stated service distance or time, given a limited (fixed) number of facilities in a network. It arises with its different variants, in several real-life problems such as police patrol planning Nodarse et al. (2019), location of Wi-Fi antennas Lee and Murray (2010), distribution of drones for medical care Pulver et al. (2016) and, location of mining bases Xue et al. (2017). It has been first introduced by Church and ReVelle (1974), where the mathematical formulation has been provided, using the following indications:
We dispose a network, with a set of nodes and arcs, defined by: $I =$ represents the set of demand nodes;
$J =$ represents the set of facility positions;
d_{ij} =the shortest distance from a node i to node j;
$S =$ denotes the coverage distance, such that a demand point i is considered "covered" if the distance from it to a given facility j is less than or equal to S ($d_{ij} \leq S$).
$N_i = \{j \in J | d_{ij} \leq S\}$. It represents the set of positions, for which the demand node i can be covered by the facility positioned in node j.
$a_i =$ population to be covered at demand node i;
$P =$ the number of facilities to be located;

The mathematical formulation is given as follows:

$$\max \quad z(y) = \sum_{i \in I} a_i \times y_i \tag{2.1}$$

$$\text{s.t.} \quad \sum_{j \in N_i} (x_j) \geq y_i, \forall i \in I \tag{2.2}$$

$$\sum_{j \in N_i} x_j = P \tag{2.3}$$

$$x_j \in \{0,1\}, y_i \in \{0,1\}, \forall i \in I, \forall j \in J, \tag{2.4}$$

Such that :

$$x_j = \begin{cases} 1 & \text{if facility is positioned in node } j \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

and,

$$y_i = \begin{cases} 1 & \text{if a demand node } i \text{ is covered by a facility within S distance} \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

The objective function 2.1 represents the total sum of the number of population served within the distance S, by the located facilities. Constraints 2.2 insure that $y_j = 1$, only when, at least, one facility is located within S distance units of the demand point i. Constraints 2.3 refers to the number of located facilities, must be equal to P.
In order to simplify the resolution of the problem, the authors proposed an equivalent mathematical formulation to the MCLP, by considering $\overline{y}_i = 1 - y_i$.

The new formulation of the MCLP is given bellow :

$$\min \quad z = \sum_{i \in I} a_i \times \overline{y}_i \tag{2.7}$$

$$\text{s.t.} \quad \sum_{j \in N_i} (x_j) + \overline{y}_i \geq 1, \forall i \in I \tag{2.8}$$

$$\sum_{j \in N_i} x_j = P \tag{2.9}$$

$$x_j \in \{0,1\}, \overline{y}_i \in \{0,1\}, \forall i \in I, \forall j \in J, \tag{2.10}$$

(For more details, see Church and ReVelle (1974).)

Herein, we show an example of maximal covering location problem, that consists on installing three short range antennas in Sao Jose Dos Campos.

2.2.1 Resolution techniques

Two techniques were proposed by Church and ReVelle (1974) to solve the maximal covering location problem (MCLP), and have tested its methods on a group of 30-nodes and 55-nodes networks instances.

2.2.1.1 Heuristic Approaches

Two heuristic approaches have been considered:
The first heuristic is called **The Greedy Adding** algorithm, (namely GA), in which we pick p

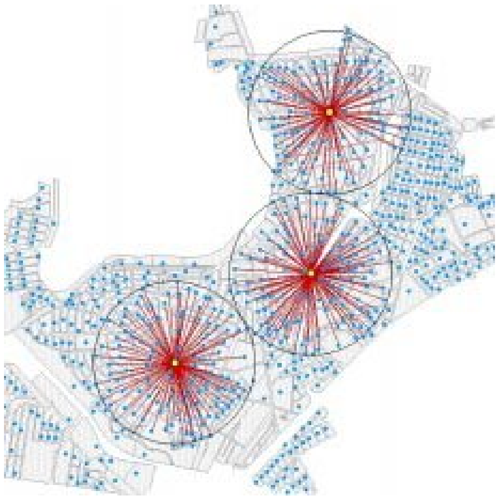

Figure 2.1: An illustrative example of MCLP

facilities sorted by their decreasing value of coverage, such that the first facility covers the most percentage of the population, the second one covers the most of the not covered population and so on.. Until, either locating p facilities or covering all the population.

The second heuristic approach is **The Greedy Adding with Substitution**, (namely GAS), which determines, as well as the GA, new locations at each iteration, but, it also improves the solution at hand, by seeking for a new better non-selected position, for each opened facility, to replace it.

2.2.1.2 Linear Programming Approach

The maximal covering location problem (MCLP), is also considered as a linear programming problem, by restricting its binary variables to a non-negative variables, which means, $x_j \geq 0$ and $y_i \geq 0$, $\forall i \in I, j \in J$.
The used approach makes sure that all x_j variables will be less than 1, otherwise it would be fixed to 1, without violating any constraints 2.8.
In addition, the minimization of the objective function provides that each $\overline{y}_i, i \in I$ will not exceed 1.
For this purpose, the solution of the linear programming can be:

1. All variables $x_j, \overline{y}_i, \forall i \in I, j \in J$ are fixed to 0 or 1.

2. Some variables are fractional (in this case, call the Branch and Bound procedure).

Meanwhile, there are other works in which, the authors proposed new techniques to tackle the basic MCLP, we can cite: Galvao and ReVelle (1996) who designed a Lagrangian heuristic. Such an approach used in the sub-gradient optimization to maximize the Lagrangian dual hoping to provide a series of tight upper and lower bounds. The experimental part analyzed the behavior of the method and compared its performance to another Lagrangian-based approach on randomly generated benchmark instances containing 150 facilities.
Therefore ReVelle et al. (2008) used heuristic concentration in order to solve the MCLP and solved it in a network with 900 nodes.

2.2.2 Large-scale Maximal Covering Location Problem

Zarandi et al. (2011) have been interested in solving the maximal covering location problem for large-scale networks, called Large-scale maximal covering location problem.
An instance of this problem is represented by a network with two identical sets of nodes and potential facilities. The objective is to find the 'optimal' locations of p facilities, for the purpose of covering the maximal amount of the population.
The mathematical formulation of the problem is given bellow: (for further explanation, readers are referred to ReVelle et al. (2008).)

$$\max \quad z(y) = \sum_{i \in I} a_i \times y_i \tag{2.11}$$

$$\text{s.t.} \quad \sum_{j \in N_i} (x_j) \geq y_i, \forall i \in I \tag{2.12}$$

$$\sum_{j \in N_i} x_j = P \tag{2.13}$$

$$0 \leq y_i \leq 1, \qquad \forall i \in I \tag{2.14}$$

$$x_j \in \{0, 1\}, \qquad \forall j \in J, \tag{2.15}$$

As it's described, the large-scale MCLP has the same mathematical formulation defined in 2.7. The only change is given in the constraints 2.14, where the use of the y variables was simplified, by considering it as a real variable in $[0, 1]$. That was justified with the first set of constraints 2.12, since they are related to the binary variables $x_j, j \in J$.

In order to tackle the large-scale MCLP, Zarandi et al. (2011) proposed a bio-inspired optimization procedure, known as: The **Genetic Algorithm** (GA). A solution of the studied MCLP, is represented by a binary vector, in which, each bit refers to the state of the node (either a facility is located inside it or not). Each solution, should exactly contain p one-value in the binary vector.
Aiming to re-product new,better solutions, the author considers a crossover and mutation operators, using the neighborhood set $N(X)$ of the solution at hand X (see Zarandi et al. (2011)).

The proposed GA was programmed, using Visual C++, in addition to CPLEX, to compare both found solutions. The tests were applied on a 1800 and 2500 nodes with different number of facilities (15,20,25).

We note that there are other papers, where nice resolution methods have been designed for several variants of the covering location (Owen and Daskin, 1998; Farahani et al., 2012) and, for more complete reviews on the domain, the reader can be referred to Snyder and Haight (2016) and Drezner and Hamacher (2002).

2.3 The Capacitated Maximal Covering Location Problem (CMCLP)

The maximal covering location problem is based on finding the best facility locations to cover the maximum number of nodes (population), and ignoring the fact that, in major realistic problem, each facility has a finite capacity of coverage or a limited distance of coverage, such as emergency services: police, ambulance, hospitals, etc.
There are many equivalent mathematical formulations to the CMCLP, however, we describe the

closest one to the MCLP formulation shown in 2.1. The following formulation is given by Pirkul and Schilling (1991):

$$
\begin{aligned}
\max \quad & z(x) = \sum_{i \in I} \sum_{j \in J} c_{ij} \times a_i \times x_{ij} && (2.16)\\
\text{s.t.} \quad & \sum_{j \in J} (y_j) \leq p && (2.17)\\
& \sum_{j \in J} x_{ij} = 1, \forall i \in I, && (2.18)\\
& x_{ij} \leq y_j, \qquad \forall i \in I, j \in J, && (2.19)\\
& \sum_{i \in I} c_{ij} \times a_{ij} \times x_{ij} \leq K_j, \qquad \forall j \in J, && (2.20)\\
& y_j \in \{0,1\}, \forall j \in J, && (2.21)\\
& x_{ij} \in \{0,1\}, \forall i \in I, j \in J, && (2.22)\\
& && (2.23)
\end{aligned}
$$

The same notation considered in 2.1 is used for the mathematical model 2.16, such that:
K_j = The capacity of the j-th facility.
d_{ij} =The distance between the the facility j to the node i.

$$c_{ij} = \begin{cases} 1, & \text{If } d_{ij} \leq S, \\ 0, & \text{Otherwise} \end{cases}$$

The introduction of the variable $c_{ij}, \forall i \in I, j \in J$,in the objective function 2.16, refers to the computing of the maximal number of demand nodes, covered by the located facilities, within their coverage area ($d_{ij} \leq S$). Constraints 2.18 insure that all demand points are assigned to a positioned facility. Constraints 2.20, guarantee that the total sum of the covered nodes by an opened facility j, should not exceed its capacity K_j.
The rest of the constraints are similar to the MCLP mathematical model. (Readers are referred to 2.1).

However, there have been many works, in which the capacitated maximal covering location problem has been studied and solved by various techniques. Herein, we are given some related works :

For the Capacitated MCLP (noted CMCLP), where the finite capacity constraint of each facility has been imposed, Pirkul and Schilling (1991) tackled it by proposing a formal description based on the strong form of the capacitated p-median problem.

Next, a Lagrangian relaxation-based solution procedure was designed for approximately solving that problem. The experimental part underlined the performance of such a method, where it was able to solve randomly generated instances containing between 50 and 100 facilities up to thirty tasks within two minutes.

A particle swarm optimization was developed by ElKady and Abdelsalam (2016) for approximately solving CMCLP. In the experimental part, the author compared the results achieved by the method to those reached by the state-of-the-art Cplex solver; the authors underlined the effectiveness of the method in determining good locations for the facility, maximizing demand coverage.

2.4 The Fuzzy Maximal Covering Location Problem (FMCLP)

One of the well-known variants of the maximal covering location problem, is the fuzzy maximal covering location problem, namely FMCLP, which arises in many realistic problems such as: the location of hospitals, ambulances, police stations, emergency units, shop and many others..

The main objective in this case, of FMCLP is to define optimal locations to install facilities, in order to serve the maximal number of demands (nodes), such that each facility has a predefined capacity of coverage.

The capacity of coverage in FMCLP is given as the distance (or travel time) of coverage, represented by two concentric circles.
The small coverage circle of each facility, has a fixed radius value R. The fuzzy distance (or fuzzy travel time) is represented by an external circle with fuzzy radius $R + f_r$.
Each demand node i, can be totally covered, partially covered, or not covered by a given facility j, according to its position, (computed as a distance or a travel time) in the coverage area of the facility j.
The literature divided the partial coverage in two classes (See Drakulic et al. (2016)). The first type concerns the MCLP variant, in which the coverage degree of a demand node is the total sum of covering degrees of partially covered locations.
A real-life example can be viewed in illumination sources. It has a radius where the light can totally be distributed to clients within it, and the light decreases, whenever the distance between the light source and the client, increases. So, the total illumination for this demand node, which is positioned in two or more external radii will be regrouped in one total sum of this partial covering degrees.
The second class regroups the problems where summing the partial covering degrees is not allowed.

In this part of study, the computing of the coverage degree $c(i,j)$ is represented by the following formula, as it has been generated in Drakulic et al. (2016), Atta et al. (2022):

$$c(i,j) = \begin{cases} 1 & \text{if } d(i,j) \leq R \\ \frac{1}{2}(\mu(x_1) + \mu(x_2)) & R < d(i,j) \leq R + f_r \\ 0 & d(i,j) > R + f_r, \end{cases} \tag{2.24}$$

Such that:
$d(i,j)$: The distance between the demand node i and the facility position j. ($i \in I, j \in J$).
$\mu(x_1)$ and $\mu(x_2)$ used by the formula (2.24) are represented in figure 2.2, as the intersections between the Right shoulder fuzzy number $R + f_r$, and the triangular fuzzy number $d_{ij} + f_d$.

The distance (or travel time) between locations, is represented The axe (OX), and in the (OY) axe, we find the coverage degrees of the locations.

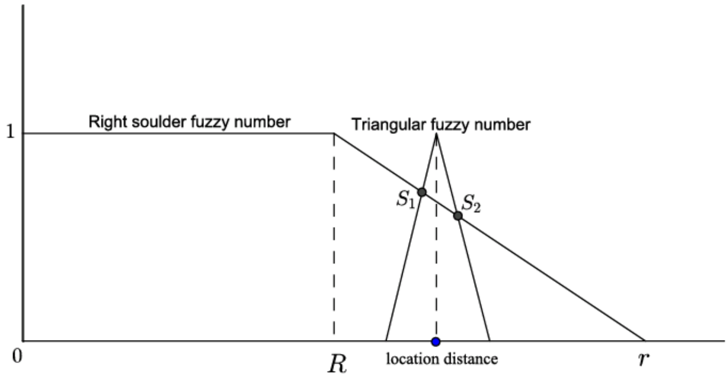

Figure 2.2: Degree of coverage

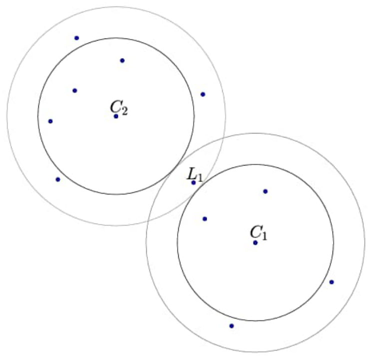

Figure 2.3: An illustrative example of FMCLP

2.4.1 Illustrative Example

In this part, we show an example about fuzzy maximal covering location problem, presented by Drakulic et al. (2016).
We dispose a set of ambulances to get installed on a given service area, such that the distance between locations are defined with two concentric circles. A demand node can be totally served if it's within the main circle, which could be represented by 5 minutes travel time radius. That can be explained as, a patient will surely have a higher chance to get a treatment at the right time. Unlike the patient who is located in the external coverage circle, with a fuzzy radius of about 10 minute travel time.
These two cases are illustrated in the following figure 2.3:

2.4.2 The mathematical formulation of FMCLP

The mathematical model of the FMCLP can be viewed as an extended version of the MCLP model, (already given in 2.1). The difference is shown in the use of the coverage degree $c(i, j)$ variables, computed in the equation 2.24, which introduce the idea of partial coverage into the MCLP model.
The definition of the FMCLP model by Drakulic et al. (2016) is given bellow:

$$\max \quad z(y) = \sum_{i \in I} y_i \tag{2.25}$$

$$\text{s.t.} \quad \sum_{j \in J} x_j = P \tag{2.26}$$

$$\max x_j \times c(i,j) \geq y_i, \quad \forall j \in J \tag{2.27}$$

$$x_j \in \{0,1\}, y_i \in [0,1], \quad \forall i \in I, \forall j \in J, \tag{2.28}$$

Such that ;
$y_i \in [0,1]$- Degree of coverage of location i.

$$x_j = \begin{cases} 1 & \text{if facility is positioned in node } j \\ 0 & \text{otherwise} \end{cases} \tag{2.29}$$

The objective of the FMCLP is to maximize the global sum of coverage, (total or partial coverage), by the set of P opened facilities.
The new constraints that have been added to the MCLP model 2.1, is the set of constraints 2.27, which refers to the maximal covering of each installed facility i.

2.4.3 Related works for solving the FMCLP

There are many related works for solving the fuzzy maximal covering location problem, for both fuzzy distance and fuzzy travel time. Herein, we give some citations of the different proposed approximately solving methods from the literature :

In Davari et al. (2013) a variable neighborhood search-based solution procedure was proposed for tackling another variant; that is,the MCLP with fuzzy coverage radii for facilities. In their computational part, the authors pinpointed the good behavior of the method, especially in instances with up to 2500 nodes. It was also able to obtain lower bounds (feasible solutions) with a gap smaller than or equal to 1.5% in much less average run-time when compared to accurate algorithms.

For the same problem, Atta et al. (2018) proposed a mimetic algorithm, in which the genetic algorithm was combined with a special refinement operator while Zarandi et al. (2011) proposed another customized genetic algorithm to solve large-scale MCLP instances, with up to 2500 facilities.

Whenever both the coverage radii of facilities and the distance between a customer and, a facility is considered fuzzy, Drakulic et al. (2016) designed a special swarm optimization for tackling it. The computational results contain two parts: (i) a first part including a comparative study between their method and the state-of-the-art Cplex solver, especially in small-sized instances containing between 70 and 90 locations, and (ii) a second part including large-scale instances, containing between 100 and 900 locations, in which the behavior of the method was studied.

Davari et al. (2011) combined the fuzzy simulation and simulated annealing for tackling the Fuzzy MCLP (FMCLP). In their experimental part, the authors studied the behavior of the method in some instances, where its performance was analyzed and affirmed that the global average objective values are less than or equal to 1.35% of the overall average optimal values.

For an extension of the fuzzy maximal covering location time problem, where travel times may be linear, trapezoidal, or Gaussian fuzzy numbers, Takaci et al. (2012) proposed an adaptation of the particle swarm optimization to solve it. In the experimental part, the authors demonstrated the effectiveness of their method when comparing its results to those published in the literature.

2.5 The fuzzy Capacitated Maximal Covering Location Problem (FCMCLP)

In this section, we tackle the most recent studied variant of the maximal covering location problem, which is the Fuzzy Capacitated Maximal Covering Location Problem (noted FCMCLP).
The FCMCLP can be viewed as a combination between the three previous problems: MCLP, CMCLP, FMCLP.
It was first introduced by Atta et al. (2022), whom proposed a multiple-metaheuristic for solving the fuzzy capacitated maximal covering location problem. The particle swarm optimization, differential evolution and two generation metaheuristics (artificial bee colony algorithm and firefly algorithm) were adapted for approximately solving the problem. In their experimental part, an extensive comparative study was conducted on 80 benchmark instances,including medium-sized and large-scale instances. Further, the results provided by all considered methods were compared, and the best achieved results were compared to those reached by the Cplex.

All the existing mathematical models of different variants of MCLP, cannot recapitulate the realistic model of the fuzzy capacitated maximal covering location problem (FCMCLP). For that purpose, Atta et al. (2022) described a new model that expresses the fuzzy distance between a customer and a facility, and the fuzzy coverage area of each facility, such that: the notion of partial coverage has been introduced, in addition to the fact that each facility has a finite supply capacity S.
The main objective of the FCMCLP is to find the sub-optimal locations to open p facilities, in order to cover the maximum percentage of total coverage degrees.

As a simple real life example of FCMCLP, we can refer to the installation of mobile towers in the area of wireless communication. Each mobile tower has a coverage radius, which depends on its signal strength that fades out with distance. The distance between a given customer and a mobile tower changes and also the capacity of service of a mobile tower is restricted to a fixed number of customers.

2.5.1 The mathematical formulation of FCMCLP

An instance of FCMCLP can be described by a set of m points on a network. Each customer $p_i, i \in \{1, ..., m\}$ has a positive demand (or weight) (w_i), and each facility $c_j, j \in \{1, ..., m\}$ has a finite supply capacity S_j. Let k be the number of facilities to be installed ($1 \leq k \leq m$). We assume that the potential location of each facility, is restricted to the customers locations (within the m points of the plane).
The following notation refers to :

R : the radius of the inner circle with center : facility location c_j.
$R + f_r$: the radius of the outer circle with center: facility location c_j.
$d(i, j)$: the Euclidean distance between the $j-th$ facility and the $i-th$ customer.
$d(i, j) \pm f_d$: the fuzzy distance between the $j-th$ facility and the $i-th$ customer.

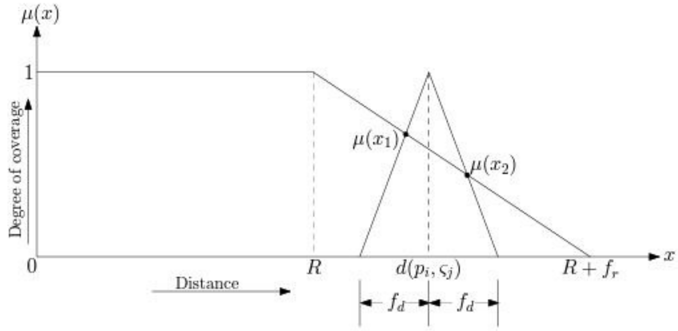

Figure 2.4: The degree of Coverage in FCMCLP

cov_{ij}: the degree of coverage of a customer i by an opened facility j, which is defined in section 2.24, has also been represented in the figure 2.4.

The FCMCLP model given by Atta et al. (2022), is illustrated bellow as :
Let $I = J = \{1, \ldots, m\}$;

$$\max \quad z(a) = \sum_{i=1}^{m}\sum_{j=1}^{m}(w_i \times cov_{ij})a_{ij} \tag{2.30}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} z_j = k \tag{2.31}$$

$$\sum_{j=1}^{m} a_{ij} \leq 1, \forall i \in I \tag{2.32}$$

$$\sum_{i=1}^{m}(w_i \times cov_{ij})a_{ij} \leq s_j z_j, \forall j \in J \tag{2.33}$$

$$a_{ij} \in \{0,1\}, z_j \in \{0,1\}, \ i \in I, j \in J, \tag{2.34}$$

Let a_{ij} be a decision variable related to linking the position p_i of the customer i and the facility c_j such that :

$$a_{ij} = \begin{cases} 1 & \text{if customer } i \text{ is served by facility } j \\ 0 & \text{otherwise} \end{cases} \tag{2.35}$$

and z_j be a decision variable related to a facility j such that

$$z_j = \begin{cases} 1 & \text{if a facility is opened at the location j} \\ 0 & \text{otherwise} \end{cases} \tag{2.36}$$

The main objective function 2.30 represents the maximum demand of covered customers. Constraints 2.31 insure that the number of facilities to be installed is equal to k. Constraints 2.32 make sure that each customer must be covered by at most one open facility. The global demand of covered customers with an installed facility, should not exceed its supply capacity, refer to 2.33.And the last constraints 2.34 are related to the binary variables used in this mathematical formulation.

2.6 Conclusion

In this chapter, we have described many kinds of variants of the well-known Maximal Covering Location Problem (namely, MCLP). Related works have been described and well resumed. We have shown the MCLP with its mathematical formulation and the different proposed resolution methods in the literature. In addition, the mathematical model of each variant of MCLP has been described and well explained. Also, we added some illustrative examples from real-life problems.

In the next chapter, we'll be interested in approximately solving the last described variant: the Fuzzy Capacitated Maximal Covering Location Problem (noted, FCMCLP).

Chapter 3

A hybrid population-based algorithm for solving the fuzzy capacitated maximal covering location problem

3.1 Introduction

The Maximal Covering Location Problem (namely MCLP) is an NP-hard combinatorial optimization problem, which arises in several real-world applications like in searching for the optimal location of emergency response facilities, services, and vehicles (Li et al., 2011), communication networks (Lee and Murray, 2010), security monitoring sensors (Murray et al., 2007) and, retail stores (Frank and Lieselot, 2007).

The well-known MCLP is based on finding the optimal location of a subset of the given facilities in order to cover the maximal number of customers positioned in a network, where each facility's coverage area and the distance between a customer and a facility, are fixed. Such constraints are not always available in reality. As an example, one can observe the constraints considered for the mobile phones network and the web-internet networks (Davari et al., 2011). Generally, a straightforward version of that problem, which also be considered as an extended MCLP with fuzzy distances, was studied and called the Fuzzy Capacitated Maximal Covering Location Problem (namely FCMCLP). The goal of FCMCLP is to establish the best opened (selected) locations regarding a fixed number of facilities that covers a maximum number of customers in a given network, such that both the distance between the customers and the installed facilities, and the coverage area of each facility are fuzzy.

Herein, we are interested in solving the Fuzzy Capacitated Maximal Covering Location Problem (FCMCLP), by using two versions of a hybrid population-based algorithm.The versions are designed as follows:(i) a modified grey wolf optimizer combined with exploration and exploitation strategies, and (ii) a modified grey wolf optimizer combined with a series of local searches and an upper bound delimiting wolf movement.

This Chapter is organized as follows. A formal description of FCMCLP is presented in Section 3.3.1. In Section 3.3, we first discuss the basic principle of the grey wolf optimizer, and next, we present the hybrid population-based algorithm designed for approximately solving FCMCLP. In the same section, an enhanced version of the algorithm is presented. Finally, in

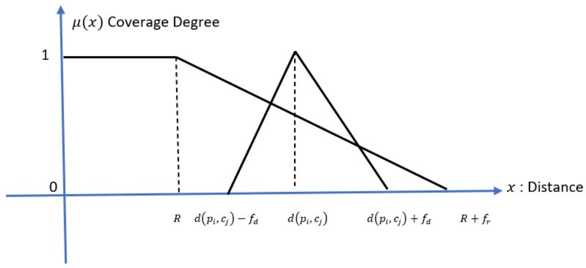

Figure 3.1: Illustration of the standard degree of coverage.

Section 3.4, the behavior of the designed algorithm is tested and analyzed for instances taken from the literature, where its achieved results are compared to those provided by a recent method of the literature and the state-of-the-art Cplex solver.

The work presented in this article has been published in an international journal, cited as follows : "Méziane Aider, Imene Dey, Mhand Hifi. "A hybrid population-based algorithm for solving the fuzzy capacitated maximal covering location problem". Computers & Industrial Engineering(2023),Volume 177,108982.DOI: https://doi.org/10.1016/j.cie.2023.108982" Aider et al. (2023).

3.2 An instance of the problem studied

In this paper, we study FCMCLP, where its instance is defined by a set P of m points (positions) in a plane and a set of k, $1 \leq k \leq m$, facilities to be installed at a chosen subset of points. Each customer $p_i \in P$, $i = 1, \ldots, m$, is represented by a non-negative demand w_i while each located facility is represented by a supply capacity s_j, $1 \leq j \leq m$. A facility can be positioned within its coverage area center c_j, which is considered as two concentric circles, with R radius and $R + f_r$ fuzzy radius, respectively. The centers of the aforementioned circles are restricted to the locations of customers themselves (Lorena and Pereira, 2002) while the coverage degree of a facility c_j positioned at a point p_i, based on the exact definition of Drakulic et al. (2016), is given as follows:

$$Cov(i,j) = \begin{cases} 1 & \text{if } d(p_i, c_j) \leq R \\ \frac{1}{2}(\mu(x_1) + \mu(x_2)) & R < d(p_i, c_j) \leq R + f_r \\ 0 & d(p_i, c_j) > R + f_r, \end{cases} \tag{3.1}$$

Where, on the one hand, $d(p_i, c_j)$ denotes the Euclidean distance between the positioned facility c_j and the customer p_i, and on the other hand, $d(p_i, c_j) \pm f_d$ denotes its fuzzy distance, such that f_d denotes a real number belonging to the interval $[5, 100]$ (Drakulic et al., 2016). Figure 3.1 illustrates both values $\mu(x_1)$ and $\mu(x_2)$ used by the formula (3.1), where the coverage degree may be computed for each used instance (as considered in Drakulic et al. (2016)).

Figure 3.2 shows a set of customers represented on the plane. On the one hand, one can observe that two facilities were positioned for covering the maximum number of overall customers. On the other hand, the same figure illustrates the two centering circles related to a given located facility, including both the radius R and the fuzzy radius $R + f_r$.

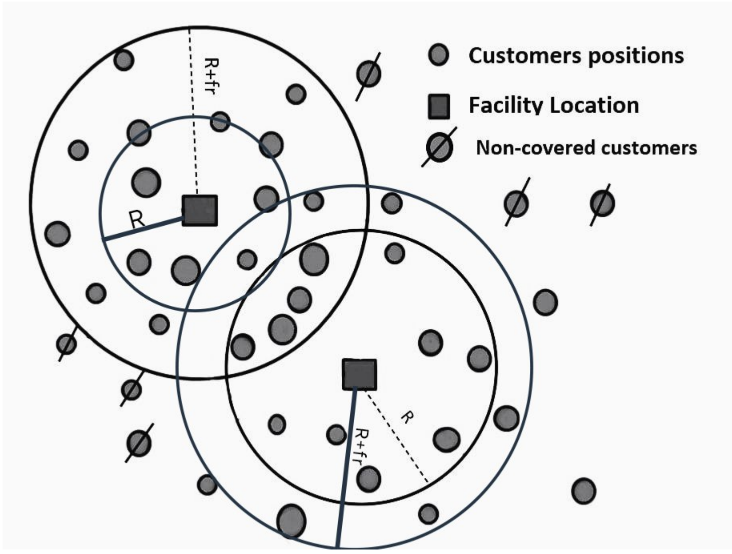

Figure 3.2: Illustration of some opened facilities coverage for a given instance.

3.2.1 Contribution

The design of powerful methods, for various combinatorial optimization problems, remains a challenge for searching solutions with high qualities. Because searching for optimal solutions remains intractable, especially when tackling "large-sized" or "arduous" instances, it is often paramount for searching good "sub-optimal" solutions for practical situations. Population-based methods, like scatter search, swarm optimization and others, have shown some advantages when adapted to different problems in this category.

In recent years, it has been observed that the hybridization of solution procedures becomes a very promising research issue. In fact, it was observed that diversified algorithms have advantages when combining them with exploration and exploitation operators. We believe that their cooperation induces a good balance between multiple- average solutions and those with high quality. Further, with the emergence of the grey wolf optimization method, several studies have been conducted to address various discrete and continuous optimization problems. Each of the studies differs, even if the goal is to design a "good" experimental method able to compete with other methods available in the literature.

Herein, we design a powerful hybrid population-based algorithm for solving FCMCLP. The designed method starts with a reference set of solutions (related to the wolves) and expands it by considering both exploration and exploitation operators. There are two versions of the method: (i) the first version adopts the principle of the discrete grey wolf optimizer hybridized with intensification and restarting search, and (ii) the second version applies the same principle as the first one except that the generation of the reference set is done around a given upper bound and the positions associated to each solution is the result of the combination between the best solutions of the reference set that is delimited by this upper bound.

The principle of the proposed method is based on the following features:

1. To build a *reference set* used as a starting population for the designed method: A first initial solution is obtained by using an iterative generator and the expanded set of solutions is built around this initial solution (ensuring the feasibility of the solutions). Moreover, the first version of the method uses a random generator for providing the starting reference set, while the second version focuses on a generation around a given upper bound.

2. Then, the following two stages are repeated:
   (a) Apply the grey wolf optimizer-based procedure for enriching the pack composed of the three best solutions and the rest of the population.
   (b) Apply a drop and rebuild operator whenever the population stagnates. In this case, the exploration of new sub-spaces is considered;which is used for diversifying the search process.
3. Steps 2a and 2b are iterated till matching the final stopping condition.

3.3 A hybrid population-based algorithm for FCMCLP

In this section, we first express the mathematical formulation of FCMCLP, already considered in the literature (Atta et al., 2022). Next, we discuss the linear relaxation of the aforementioned model for establishing an upper bound for the hybrid method. Further, we describe the basic principle of the grey wolf optimizer, including its main steps. Finally, we discuss how such a method can be adapted (and enhanced) for tackling the FCMCLP.

3.3.1 FCMCLP

Let x_{ij} be a decision variable related to linking the position p_i of the customer i and the facility c_j such that :

$$x_{ij} = \begin{cases} 1 & \text{if customer } i \text{ is served by facility } j \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

and y_j be a decision variable related to a facility j such that

$$y_j = \begin{cases} 1 & \text{if a facility is opened at the location j} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

Let $I = J = \{1, \ldots, m\}$; then, a formal description of FCMCLP may be stated as follows:

$$\max \quad z(x) = \sum_{i=1}^{m}\sum_{j=1}^{m}(w_i \times cov_{ij})x_{ij} \tag{3.4}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} y_j = k \tag{3.5}$$

$$\sum_{j=1}^{m} x_{ij} \leq 1, \forall i \in I \tag{3.6}$$

$$\sum_{i=1}^{m}(w_i \times cov_{ij})x_{ij} \leq s_j y_j, \forall j \in J \tag{3.7}$$

$$x_{ij} \in \{0,1\}, y_j \in \{0,1\}, \; i \in I, j \in J, \tag{3.8}$$

where equation (3.4) denotes the objective function that maximizes the sum of demands satisfied by the opened (positioned / selected) facilities according to their coverage degrees. The constraints (3.5) make sure that only k facilities have been positioned in the network. Constraints (3.6) verify if each customer's demand is satisfied by at most one opened facility. Each inequality (3.7) requires that the global demand satisfied by the facility j is at most equal to the total capacity of the facility j, which explains that a given facility j can partially cover a given customer i with a certain coverage degree cov_{ij}, and the sum of the whole demands of the partially(or completely) covered customers by this facility are within its capacity s_j . Finally, the last line (3.8) of the model is related to the domain of the decision variables.

We have already mentioned the need of an upper bound in order to make the solutions of the referent set cooperate with this bound. Let $\mathrm{RP}_{\mathrm{FCMCLP}}$ be the resulting program induced from FCMCLP by relaxing overall binary variables, i.e., setting $x_{ij} \in [0,1]$, and $y_j \in [0,1], \forall i \in I, \forall j \in J$ such that $\mathrm{RP}_{\mathrm{FCMCLP}}$ is given as follows:

$$\begin{aligned}
\max \quad z(x) = \quad & \sum_{i=1}^{m}\sum_{j=1}^{m}(w_i \times cov_{ij})x_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{m} y_j = k \\
& \sum_{j=1}^{m} x_{ij} \leq 1, \forall i \in I \\
& \sum_{i=1}^{m}(w_i \times cov_{ij})x_{ij} \leq s_j y_j, \forall j \in J \\
& x_{ij} \in [0,1], y_j \in [0,1],\ i \in I, j \in J,
\end{aligned} \tag{3.9}$$

The relaxation $\mathrm{RP}_{\mathrm{FCMCLP}}$ is optimized by using the Simplex method. Then, we collect the current primal solution representing the primal configuration; that is an upper bound for the original problem FCMCLP. Of course, one can observe that relaxing only the decision variables x_{ij} may provide a tighter upper bound. However, limited computational results showed that relaxing all variables provided an upper bound very close to the latter, but required much more runtime, especially for large-sized instances. Hence, we limit this study to the bound produced by the Simplex method.

3.3.2 A basic grey wolf optimizer

The Grey Wolf Optimizer (GWO) is a recent meta-heuristic inspired from the grey wolves in nature. Often, such an approach tries to mimic the leadership hierarchy and hunting mechanism of grey wolves, and to use it as a search process for generating series of positions related to feasible solutions to a given problem. The grey wolf lives in groups limited to a small number of wolves, which contains the leader wolf, noted α, who drives the hunting. A standard hierarchy leadership has the wolf, noted β, on its second level, which is the next best wolf in the pack, and then it follows the third level known as the wolf, noted δ. The lowest ranking grey wolves represent a subgroup of the rest of the wolves, noted ω each, where each of them respects the decisions provided by the three dominant wolves, i.e., α, β and δ.

In addition to the social hierarchy of wolves, group hunting is another interesting social behavior of grey wolves. According to Muro et al. (2011), the main phases of grey wolf hunting are represented by tracking, encircling, and attacking the prey. Formulas related to the social hierarchy with the main phases of hunting are given in what follows, such that the wolves use a predefined direction to encircle the prey, where each wolf's position is updated according to the prey's position:

$$\overrightarrow{D} = |\overrightarrow{C} \cdot \overrightarrow{S_p}(t) - \overrightarrow{S}(t)| \tag{3.10}$$

where the new position related to each wolf will be updated as follows:

$$\overrightarrow{S}(t+1) = \overrightarrow{S_p} - \overrightarrow{A} \cdot \overrightarrow{D}, \tag{3.11}$$

which aims to encircle the prey. In this case, for equation (3.11), t denotes the current iteration, $\overrightarrow{S}(t+1)$ is the position vector of the wolf, $\overrightarrow{S_p}$ denotes the position vector of the prey, while $\overrightarrow{A}$

and $\overrightarrow{C}$ are two coefficient vectors defined as follows:

$$\overrightarrow{A} = 2\overrightarrow{a} \cdot \overrightarrow{r_1} - \overrightarrow{a} \quad \text{and} \quad \overrightarrow{C} = 2\overrightarrow{r_2} \tag{3.12}$$

On the one hand, $\overrightarrow{a}$ (used in the first equation of (3.12), on the left-hand) is a vector initialized to 2 and decreases till matching the value 0 throughout the iterations; in this case, we say that the method converges to final position if that method is well-tailored for the studied problem. On the other hand, $\overrightarrow{r_1}$ (on in the first equation of (3.12), on the left-hand) and $\overrightarrow{r_2}$ (on the second equation of (3.12), on the right-hand) denote two vectors whose components are randomly generated in the continuous interval $[0, 1]$.

Because the prey's position is generally non-obvious to compute, especially when tackling NP-Hard combinatorial optimization problems containing several local optima, we then simulate such a position according to the best position related to the leader of the pack. Hence, at each iteration of the search process, simulating the positions of the grey wolf hunting may be based upon the three best solutions α, β and δ, and by forcing the other obtained wolves to update their positions according to the position of the best search solution, updated with the following formulas:

$$\overrightarrow{D_\lambda} = |\overrightarrow{C_\lambda} \cdot \overrightarrow{S_\lambda}(t) - \overrightarrow{S}(t)|, \tag{3.13}$$

where λ represents the α, β and δ wolves by affecting their corresponding directions $\overrightarrow{D_\alpha}, \overrightarrow{D_\beta}$ and $\overrightarrow{D_\delta}$ respectively. Differently stated, the three best positions will be updated as follows:

$$\overrightarrow{S_1} = \overrightarrow{S^{(\alpha)}} - \overrightarrow{A_1} \cdot \overrightarrow{D_\alpha}, \quad \overrightarrow{S_2} = \overrightarrow{S^{(\beta)}} - \overrightarrow{A_2} \cdot \overrightarrow{D_\beta} \quad \text{and} \quad \overrightarrow{S_3} = \overrightarrow{S^{(\delta)}} - \overrightarrow{A_3} \cdot \overrightarrow{D_\delta}. \tag{3.14}$$

Finally, for the next iteration $(t + 1)$, each new position related to the current wolf is updated as follows:

$$\overrightarrow{S}(t+1) = \frac{\overrightarrow{S_1} + \overrightarrow{S_2} + \overrightarrow{S_3}}{3}. \tag{3.15}$$

In order to make the paper self-containing, we consider in what follows that $S^{(i)}$ as the solution / configuration of the wolf i in the pack, and $S_1^{(i)}$, $S_2^{(i)}$ and $S_3^{(i)}$ are the computed solutions / configurations when using the three best wolves for computing the new solution / configuration associated to the wolf i.

Input. A starting population $\mathcal{P}$ of size $|\mathcal{P}| > 0$.
Output. $S^{(\alpha)}$: the best position of the final population.

Let $\mathcal{P}$ be the initial population of positions $S^{(i)}$, $i = 1, \dots, |\mathcal{P}|$.
Evaluate the fitness function z_i related to each wolf, i.e., $z(S^{(i)})$ related to the position $S^{(i)}$, $i = 1, \dots, |\mathcal{P}|$.
Let $S^{(\alpha)}$, $S^{(\beta)}$ and $S^{(\delta)}$ be the three best positions of $\mathcal{P}$ according to their finesses.
Set $t = 0$ and let t_{max} be a maximum number of iterations.
Initialize a, A and C.
while ($(a \neq 0)$ and $(t < t_{max})$) **do**
  for ($i = 1$ (position of the first wolf of $\mathcal{P}$) to $|\mathcal{P}|$) **do**
    Compute S_1, S_2 and S_3 (according to Eq (3.14)).
    Evaluate the new position $S^{(i)}_{new}$ of the current wolf (according to Eq (3.15)).
    if ($z(S^{(i)}_{new}) > z(S^{(i)})$) **then**
      Update $S^{(i)}$ with $S^{(i)}_{new}$.
    end if
  end for
  Update the three best positions of wolves $S^{(\alpha)}$, $S^{(\beta)}$ and $S^{(\delta)}$, according to their fitness values.
  Set $a = (2 - t \times \frac{2}{t_{max}})$ and update A and C according to Eq (3.12).
  Increment t.
end while

Algorithm 2: - A standard Grey Wolf Optimizer (GWO)

3.3.2.1 A hybrid algorithm for FCMCLP

GWO works with a group of wolves, which can be viewed as a population-based method. Hence, GWO may first start with a randomly generated population $\mathcal{P}$ of wolves (solutions), using a generator trying to cover the search space of the problem concerned for diversification purposes (Section 3.3.2.2). Second, at each step, regarding the position of the prey, the positions of the wolves should be computed regarding their positions and the positions of the three best wolfs' positions in the population (Section 3.3.2.3). Third, the new population is ranked in decreasing order of their finesses (a maximization problem). Fourth and last, such a process is iterated until matching the stopping criteria. Algorithm 2 describes the main steps of GWO-based population method. In what follows, we mimic such a process for achieving a near-optimal solution for FCMCLP.

3.3.2.2 A starting population / reference set

Because two versions of the hybrid population-based algorithm are proposed, we then describe how the reference set can be generated. GWO can be viewed as a stochastic method, we then randomly generate a set $\mathcal{P}$ of feasible solutions, where each solution is assigned to a wolf's position.

First version. The starting population $\mathcal{P}$, for the first version of the algorithm, is generated by using the following *Starting Generator Procedure* (SGP), where (In what follows, we consider $\overline{S}^{(i)}$ as the i-th wolf / solution):

Second version. The starting population $\mathcal{P}$, for the second version of the algorithm, is generated by cooperating a random generator with the optimal solution of $\mathrm{RP}_{\mathrm{FCMCLP}}$, provided

Inputs. Let $\rho = |\mathcal{P}|$ (population size), and m be the number of customers .
Output. A reference set (starting population) for FCMCLP.

Set $\mathcal{P} = \emptyset$ (by setting the population size to $\rho > 0$).
Set $i = 0$ and $nb_{\text{Iter}} = 0$ (the maximum number of iterations is fixed to Max-nb_{Iter}).
repeat
  $i = i + 1$ and set $\overline{S}^{(i)} = \emptyset$.
  repeat
    Set $\ell =$`rand`$(1, m)$.
    if $(\overline{S}^{(i)} \cup \ell$ is feasible) **then**
      update $\overline{S}^{(i)} = \overline{S}^{(i)} \cup \ell$.
    else
      close (remove) it from $\overline{S}^{(i)}$.
    end if
  until (providing a complete solution).
  if $(\overline{S}^{(i)} \notin \mathcal{P})$ **then**
    set $\mathcal{P} = \mathcal{P} \cup \overline{S}^{(i)}$
  else
    Decrement(i).
  end if
  Increment(nb_{Iter}).
until $(i = \rho$ or $nb_{\text{Iter}} =$Max-$nb_{\text{Iter}})$.

Algorithm 3: - Starting population Generator

with the Simplex method. The reference set is reached by using a modified SGP, namely SGP_U, described in what follows:

On the one hand, one can observe that the above procedures (Algorithms 3 and 4) try to generate ρ feasible solutions by applying two loops. The external loop may be modified whenever the ρ solutions are not matched (by adding a maximum number of iterations) and so, the population may be completed when calling the core of the method. On the other hand, we can remark the difference between both procedures is related to the use of an upper bound in Algorithms 4 (line 3), especially fixing the k largest fractional values that the Simplex provides; in this case, the Simplex method often favors these variables to optimize the problem instance, hence the strategy is related to fixing these greatest values.

3.3.2.3 Generating new positions

A *Generational Procedure* (GP) is related to the successive positions induced by overall wolves during the search process. Determining a new position of each wolf is equivalent to combine the current wolf's position and the positions related to the three best positions (solutions) of the population. We recall that equation (3.15) often induces a fractional solution whenever at least one component, at the same rank of each solution, was fixed to one, i.e., the facility is open at this position. Such a formula tries to compute the new wolf's position, which depends on the position of the virtual prey; Which is simulated (replaced) to the combination of the three best positions. Herein, we make a cooperation between a series of fractional positions, where each i-th integral position (solution) of the population $\mathcal{P}$ may be provided by using the following

Inputs. $(\bar{x}, \bar{y})$ ($\mathrm{RP}_{\mathrm{FCMCLP}}$'s optimal solution), $\rho = |\mathcal{P}|$ (the population size), and m (the number of customers).
Output. A reference set (starting population) for FCMCLP.

1: Set $\mathcal{P} = \emptyset$ (by setting the population size to $\rho > 0$).
2: Set $i = 0$ and $nb_{\mathrm{Iter}} = 0$ (the maximum number of iterations is fixed to Max-nb_{Iter}).
3: `// Generating the first reference solution around the upper bound`
- Order the vector $\bar{y}$ in decreasing order of their fractional values.
- Fix the k first $\bar{y}_j$ to 1, $j \in J$, and let $\hat{S}$ be the current partial solution.
4: **repeat**
5: $\quad i = i + 1$ and set $\overline{S}^{(i)} = \hat{S}$.
6: $\quad$ **repeat**
7: $\quad\quad$ Set $\ell =$`rand`$(1, m)$.
8: $\quad\quad$ **if** $(\overline{S}^{(i)} \cup \ell$ is feasible) **then**
9: $\quad\quad\quad$ update $\overline{S}^{(i)} = \overline{S}^{(i)} \cup \ell$.
10: $\quad\quad$ **else**
11: $\quad\quad\quad$ close (remove) it from $\overline{S}^{(i)}$.
12: $\quad\quad$ **end if**
13: $\quad$ **until** (providing a complete solution around $\hat{S}$).
14: $\quad$ **if** $(\overline{S}^{(i)} \notin \mathcal{P})$ **then**
15: $\quad\quad$ set $\mathcal{P} = \mathcal{P} \cup \overline{S}^{(i)}$
16: $\quad$ **else**
17: $\quad\quad$ Decrement(i) .
18: $\quad$ **end if**
19: $\quad$ Increment(nb_{Iter}).
20: **until** ($i = \rho$ or $nb_{\mathrm{Iter}} =$Max-nb_{Iter}).

Algorithm 4: - Starting population Generator using an upper bound

transformation according to the new achieved position:

1. Let f be the sigmoid function assigned to each element $\overline{S}_j^{(i)}$ of the solution $\overline{S}^{(i)}$, defined as follows:
$$f(\overline{S}_j^{(i)}) = \frac{1}{1 + e^{-\overline{S}_j^{(i)}}} \tag{3.16}$$

2. Check if $f(\overline{S}_j^{(i)}) \leq \frac{1}{2}$ then set the j-th component of the solution $\overline{S}^{(i)}$ to $\left\lceil \overline{S}_j^{(i)} \right\rceil$, set it to $\left\lfloor \overline{S}_j^{(i)} \right\rfloor$ otherwise.

We note that $\lceil x \rceil$ (resp. $\lfloor x \rfloor$) is the greatest (resp. the lowest) integer value related to x.

Now, one can observe that each component $\overline{S}_j^{(i)}$ may provide a value out of the discrete interval $\{1, \ldots, n\}$ (the greatest(respectively the lowest) bound exceed n (respectively less than 1), where that value reflects the assignment of an opened facility in a certain point p_j. Hence, in order to work with feasibility, we propose the following greedy operator to correct some of these components; That is composed of two steps:

Therefore, the procedure makes sure to provide a new feasible solution, by verifying at each

Inputs. Let $\overline{S}^{(i)}$ be the current integral position provided by applying Eq (3.16).
Output. A feasible solution $\overline{S}^{(i)}$.

1: Suppose that all components of $\overline{S}^{(i)}$ are unfixed (free).
2: Let $\delta_k = \sum_{\substack{j=1 \\ j \neq k}}^{n} cov(k,j)$ be the degree of the facility k.
3: **repeat**
4: Set $\frac{w_k^{(i)}}{\delta_k^{(i)}}$, $k = 1, \ldots, n$, in decreasing order, where $\delta_k^{(i)}$ (resp. w_k) denotes the degree (resp. weight) of the facility (resp. the customer) k of the i-th configuration.
5: Set $\ell = \arg \max_{1 \leq k \leq n} \left\{ \frac{w_k^{(i)}}{\delta_k^{(i)}} \right\}$.
6: Check the feasibility by fixing the facility ℓ (that does not already exist in the solution $\overline{S}^{(i)}$), to close it otherwise.
7: **until** providing a complete solution (fixing −closed or open−all possible facilities).

Algorithm 5: - Greedy operator

step of including a new opened facility that does not appear in the set $\overline{S}^{(i)}$ (which contains only the current opened facilities). A call for a simple loop is given as follows:

1. Let j be an opened facility in $\overline{S}^{(i)}$.

2. Check the number of occurrences (set it to $occur^{(i)}(j)$) of j in $\overline{S}^{(i)}$.

3. **Do**
   - Replace all the occurrences of j with ℓ, such that $\ell = \text{rand}[1, .., m]$ and $\ell \neq j$.
   While $(occur^{(i)}(j) > 1)$.

3.3.3 Intensification search

Searching for an improved solution (integral position) is equivalent to fix some stations among the n possible stations. Making some moves between fixed stations is equivalent to fix some of them and to close the rest of the facilities.

3.3.3.1 A 2-opt operator

Such an operator can be viewed as a simple local search, which is even based upon basic local modifications of the solution at hand. Often, the operator sequentially makes a series of swaps as long as the quality of the induced solution is enhanced. Whenever the solution stagnates, where its objective value remains quasi-unchanged, then we may estimate that the operator converges to a local optimum. Herein, we propose to swap between a current opened facility k of the i-th

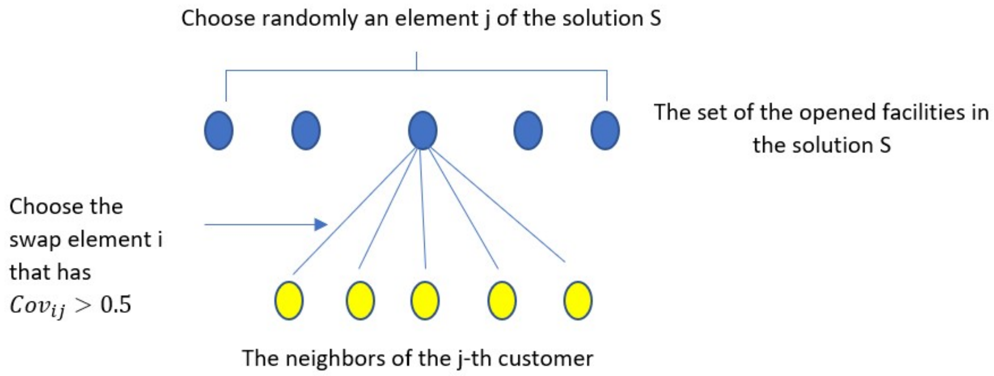

Figure 3.3: Simple two permutation for a FCMCLP feasible solution

solution $\overline{S}^{(i)}$ with a chosen facility satisfying the following criteria:

$$\ell = \arg \max_{\substack{1 \leq r \leq n: \\ r = \text{unfixed}}} \Big\{ Cov(k, r),\ r \in \{1, \dots, n\}, \\ k \neq r,\ k = \text{fixed} \Big\}. \tag{3.17}$$

Of course, for each new opened facility ℓ, the procedure checks the feasibility related to the induced solution.

Figure 3.3 illustrates how the 2-opt operator works, regarding a given solution $\overline{S}$. Of course, we can change the condition related to the swapping, between a random element of the solution (which refers to a certain positioned facility) with one of its neighbors. We prefer to do the 2 swap with a new neighbor that has a better coverage degree. The same figure refers to a permutation between the facility j with its neighbor i that has the value related to cov_{ij} greater or equal to $\frac{1}{2}$. Finally, such an operator can be applied to a subgroup of located facilities of the solution $\overline{S}$, and then check the feasibility of the new achieved solution.

3.3.3.2 Avoid cycling

Often, the 2-opt operator tries to build a series of solutions around other solutions. A subset of solutions may be viewed as a neighborhood, according to a given solution. Because a new solution built can be provided by swapping two facilities, one can observe that repeating the same process may lead toward the same local optimum and so, the operator may provide a premature convergence towards local optimum. There are several techniques that can be employed to avoid cycling towards the same solutions. In our study, we considered the 2-opt operator combined with a tabu list in order to mimic the well-known tabu search (Glover, 1989), where the tabu list is characterized by a list of temporary reverse swaps that avoids to return to the solutions already visited.

Because we limit the size of each tabu list, we also used a hashing function, where a list of objective functions is stored during the search process. In this case, a list of Θ bounds (related to the best fitness values) is stored, where each bound may correspond to different solutions. Of course, the smaller the value of Θ is, the more refined results can be achieved, but in this case, it will often be at the expense of the run time. Our limited computational results showed that setting Θ to 20 provided a good compromise between solution quality and the average run time needed by the method.

Diversification search

Often, the intensification search tries to enhance the solutions at hand. Differently stated, its goal is to provide a series of neighborhoods, issuing from a series of given solutions, which might contain solutions with better quality. Despite some enhancements that can be achieved by such a strategy, and because of the number of achievable solutions with the same objective value, it is interesting to provide a manner able to drive the search process through other unvisited sub-spaces. In order to simulate such a diversified operator, we propose to add the Drop and Rebuild Operator (DRO) already used in Hifi (2002), where it aims at removing a part of assigned facilities from the current solution. Such a strategy tries to diversify the search process by degrading the quality of the solution at hand hoping to avoid the stagnation in the local optima.

In this case, a first partial (incomplete) solution is built; that is completed by using a part of the procedure described in Section 3.3.2.3. It can be summarized as follows:

1. Let $\overline{S}^{(i)}$ be the current solution.
2. Drop $\beta\%$ of the fixed facilities belonging to $\overline{S}^{(i)}$.
3. Complete the partial solution by adding the facilities satisfying the best weight per degree, i.e., each added facility ℓ must satisfy

$$\ell = \arg \max_{\substack{1\leq k\leq n,\\ k=\text{non-fixed}}} \left\{\frac{w_k^{(i)}}{\delta_k^{(i)}}\right\}.$$

3.3.4 An overview of the proposed hybrid population-based algorithm

Algorithm 6 summarizes the main steps of the proposed Hybrid Population-Based Algorithm (HP-BA). It starts (line 1) with a population $\mathcal{P}$ of solutions (related to the wolves) reached by applying one of the constructive procedures discussed in Section 3.3.2.2. Line 2 ensures the feasibility of the overall solutions of $\mathcal{P}$ by applying the greedy procedure GP (cf., Section 3.3.2.3). Line 3 stores the best current solution, related the wolf α; That is $S^\star$ whose objective value realizes $z(S^\star) = \max_{1\leq i\leq |P|}\{z_i(S^{(i)}) \mid S^{(i)} \in P\}$. Next, the algorithms uses two loops: (i) an *internal loop while* (from line 8 to line 20) and (b) the *global loop repeat* which participates to the diversification of the population for jumping from a region to another non-explored region of the decision space. which applies the GWO for enriching the pack $\mathcal{P}$. Of course, the diversification strategy is that described in Section 3.3.3.2, where DRO is applied. Finally, the search process continues till matching the global stopping criteria, where the algorithm exists with the best solution $S^\star$.

3.4 Computational results

The objective of this part is to assess the behavior of the designed Hybrid Population-Based Algorithm (HP-BA) by comparing its achieved results to those provided by more recent and competitive methods available on the literature and the state-of-the-art Cplex solver. All considered methods were evaluated on the same set of instances extracted from Atta et al. (2022), where each set contains between seven and fifteen values of k, denoting the number of facilities to open, and overall groups are differentiated with the values assigned to m (as discussed below, and displayed in table 3.1).

The first column of Table 3.1 represents the different values related to m; that is the number of customers to be served. We note that m belongs to the discrete interval $\{324, 402, 500, 708, 818, 700, 900\}$.

Input. An instance of FCMCLP.
Output. The best solution $S^{\star}$ for FCMCLP.

Let $\mathcal{P}$ be the starting reference set of wolfs built by calling either SGP or SGP_U (cf., Section 3.3.2.2).
Apply the generational procedure **??** to $\mathcal{P}$ to ensure the feasibility of overall positions $S^{(i)}$, $i = 1, \ldots, |\mathcal{P}|$ (cf., Section 3.3.2.3).
Set $S^{\star}$ as the best solution of $\mathcal{P}$ such that $z(S^{\star}) = \max_{1 \le i \le |\mathcal{P}|} \{z_i(S^{(i)}) \mid S^{(i)} \in \mathcal{P}\}$.
Let t_{max} be a maximum number of iterations.
repeat
  Let $X^{(\alpha)}$, $X^{(\beta)}$ and $X^{(\delta)}$ be the three best positions of $\mathcal{P}$ according to their finesses.
  Set $t = 0$ and initialize a, A and C.
  while ($(a \neq 0)$ and $(t < t_{max})$) **do**
    for ($i = 1$ (position if the first wolf of $\mathcal{P}$) to $|\mathcal{P}|$) **do**
      Let $X^{(i)}$ be the current solution of $\mathcal{P}$.
      Compute $X_1^{(i)}, X_2^{(i)}$ and $X_3^{(i)}$
      (according to Eq (3.14)), and $X_{new}^{(i)}$ (according to Eq (3.15)).
      Check (and improve) the feasibility of the new position $X_{new}^{(i)}$ by calling the 2-opt operator with non-cycling strategy (cf. Sections 3.3.3.1 and 3.3.3.2).
      if ($z(X_{new}^{(i)}) > z(X^{(i)})$) **then**
        Update $X^{(i)}$ with $X_{new}^{(i)}$.
      end if
    end for
    Set $a = (2 - t \times \dfrac{2}{t_{max}})$ and update A and C according to Eq (3.12) .
    Update the three best positions of wolves $X^{(\alpha)}$, $X^{(\beta)}$ and $X^{(\delta)}$ according to their fitnesses.
    Update the best solution $S^{\star}$ and increment t.
  end while
  Apply the diversification strategy on the final population $\mathcal{P}$ by calling the drop and rebuild operator DRO (cf., Section 3.3.3.2).
until (satisfying the global stopping criteria)
Exit with the best solution $S^{\star}$.

Algorithm 6: - A Hybrid Population-Based Algorithm (HP-BA) for FCMCLP

The second column refers to the number of the opened facilities k, as discussed in 3.1, where R represents the radius of the coverage area of circles of a given facility, $R + f_r$ denotes the the fuzzy radius, while f_d is the fuzzy added value to the distance between each customer and installed facility. As underlined in the same table, their values are fixed either to $600/300/100$ or $20/10/5$. The last column shows the supply capacity of any opened facility in the network, which is either equals to 4500 or 9500.

Table 3.1: Characteristics of the instances

m	k	$R/f_r/f_d$	S
324	$\{1,7\}$	600/300/100	4500
402	$\{1,9\}$	600/300/100	4500
500	$\{1,13\}$	600/300/100	4500
708	$\{1,15\}$	600/300/100	4500
818	$\{1,15\}$	600/300/100	4500
700	$\{1,12\}$	20/10/5	9500
900	$\{1,13\}$	20/10/5	9500

3.4.1 Behavior of the first version of HP-BA

3.4.1.1 Parameter settings

Often, a general purpose heuristic may lead results of variable quality, which is related to a variety of parameters used for tackling large-scale optimization problems. Further, the proposed HP-BA needs three decision parameters: (i) the size of the population, (ii) the number of items to drop for using a re-optimization procedure, and (iii) the criterion used for stopping the search process.

3.4.1.2 A. Effect of the population size

In the preliminary study, the behavior of the first version of HP-BA (without using the upper bound) was tested on the first 28 instances contained in the six groups. Its behavior was evaluated by varying the size of the population, where initially that population was fixed to the minimum size of 5 and doubled for the next version. For each run, the run time limit was fixed between ten and thirty minutes (according to the size of the instance).

Table 3.2: Behavior of HP-BA when varying the population size Pop_{size}

		%Cov: solution values			Runtime(min)			
m	k	HP-BA$_5$	HP-BA$_{10}$	HP-BA$_{20}$	HP-BA$_5$	HP-BA$_{10}$	HP-BA$_{20}$	Best lit results
324	1	**37.03**	**37.03**	**37.03**	1	1	1	**37.03**
	2	63.73	**65.14**	62.93	2	2	2	65.14
	5	98.1	**99.07**	95.77	4	10	6	98.82
402	2	56.04	**56.10**	56.04	6	6	7	56.10
	3	**77.25**	**77.25**	77.25	6	8	9	77.25
500	1	**22.83**	**22.83**	**22.83**	1	1	1	22.83
	4	77.8	**80.19**	78.92	20	4	20	79.94
708	1	18.599	**18.6**	18.592	2	4	5	**18.6**
	2	37.14	**37.2**	37.15	6	7	7	**37.2**
700	5	83.67	**83.8**	83.65	30	25	30	83.36
	6	88.46	88.56	**89.09**	11	8	10	87.44
900	2	**26.35**	**26.35**	**26.35**	3	3	3	**26.35**
	4	**56.7**	**56.7**	**56.7**	8	9	10	52.7
Average		57.30	**57.71**	57.15	7.69	**6.77**	8.54	57.14

Table 3.2 reports HP-BAs' results, where columns 1 and 2 display the instance information, columns from 3 to 5 show the best upper bounds (percentage coverage, noted HP-BA$_5$, HP-BA$_{10}$ and HP-BA$_{20}$, respectively) achieved by HP-BA according to overall values related to the size of the population, and column 6 to column 8 report HP-BA's run time when varying the population size. Finally, the last line of the table tallies the average lower bound of all tested

instances for each version of HP-BA and, the average run time consumed by HP-BA$_5$, HP-BA$_{10}$ and HP-BA$_{20}$, respectively. In what follows, we comment on the results of Table 3.2:

- For all considered population sizes, all versions of HP-BA are able to match several best bounds achieved by the more recent and the best algorithms available in the literature (extracted from Atta et al. (2022)) according to the size of the instance.
- For overall instances, HP-BA$_{10}$ version (with a population size $\text{Pop}_{\text{size}} = 10$) reaches a better behavior when comparing its value to those reached by other versions of the algorithm. Indeed, HP-BA$_{10}$ achieves an average value of 57.71, which realizes a Gap (HP-BA$_{10}$-HP-BA$_5$, with a population size $\text{Pop}_{\text{size}} = 5$) of 0.41 when compared to the average value related to the second best average bounds of HP-BA.
- HP-BA$_{10}$ seems slightly faster than the two other versions of HP-BA; in this case, HP-BA needs an average run time of 6.77 minutes while HP-BA$_5$ (resp. HP-BA$_{20}$ with a population size $\text{Pop}_{\text{size}} = 20$) consumes 7.69 (resp. 8.54) minutes.

Because we are looking for the version that is able to achieve solutions with high quality (greatest objective values) and with more best bounds, we then fix $\text{Pop}_{\text{size}} = 10$ for the rest of the computational results.

3.4.1.3 B. Effect of the drop and rebuild operator

In this part, we discuss the effect of the diversification strategy that is based upon the drop and rebuild operator (noted DRO in Section 3.3.3.2). We recall that DRO consist in removing a subset of located facilities, which induces a partial solution and, completed it with other facilities by checking the feasibility. Removing located facilities is equivalent to remove β% of these facilities. Hence, the behavior of the method is evaluated on subset of instances, where the run time limit was fixed according to the size of the instance tested (herein, twenty minutes are considered for medium instance and thirty minutes for large-sized ones).

Table 3.3: Effect of HP-BA when using DRO

		Variation of the parameter β					
m	k	5%	10%	15%	30%	50%	60%
324	2	64.19	**65.14**	65.20	64.82	65.20	65.20
	5	99.09	96.56	97.62	**99.10**	**99.10**	97.20
	6	98.33	98.64	**99.84**	99.03	**99.84**	99.18
700	2	**34.02**	**34.02**	**34.02**	**34.02**	**34.02**	**34.02**
	4	68.02	68.02	**68.03**	68.02	**68.03**	**68.03**
	7	90.47	90.57	90.57	90.91	**91.32**	90.91
500	2	45.64	45.60	45.61	45.61	**45.66**	45.63
	3	66.01	**66.44**	65.00	66.28	**66.45**	66.28
818	4	54.74	**55.85**	55.62	59.64	59.64	59.64
	5	65.03	65.03	66.30	65.98	**72.00**	68.00
Average		68.55	68.62	68.78	69.34	**70.13**	69.41

Table 3.3 reports the bounds achieved by HP-BA when varying the value of β in the discrete interval {5%, 10%, 15%, 30%, 50%, 60%}. Columns 1 and 2 of the table refer to the instance's information, columns from 3 to 8, under the value assigned to β, display for each instance both the average lower bound achieved by the method over the ten trials. Finally, the last lines summarize the global average bounds achieved by the method. From Table 3.3, we observe what follows:

- The version with $\beta = 50\%$ (column 7 and the last line, in bold-space) is able to achieve a better global average value of 70.13, when compared to the other versions of the method. With $\beta = 50\%$, the method matches 8 average bounds over the ten instances tested, which represents a percentage of 80%.
- The last version with $\beta = 60\%$ achieves the second best global average bound of 69.41 and it matches 2 better values while the rest of the versions reach a global average value varying from 68.55 (with $\beta = 5\%$) to 69.34 (with $\beta = 30\%$).
- Of course, even the average bound reached by the method with $\beta = 60\%$ remains interesting, but it is still far from the value achieved by the method with $\beta = 50\%$. Indeed, the best version realizes a gap of 0.72, which remains quite consequent for such a problem.

For the rest of the experimental part, and before fixing the value of β, we provide a statistical analysis on the average lower bounds achieved by all versions of the first version of HP-BA, where the parameter β is considered variable as above.

C. Statistical analysis

We then apply two well-known statistical measures: (i) the *sign test* which computes both the number of non-negative (> 0) and negative (< 0) gaps (comparing the achieved gaps related to the difference between two bounds provided by two versions of HP-BA) and, (ii) the *Wilcoxon signed-rank test* statistics that is considered as an alternative study expressing the p-value for indicating how one version performs better than another. We do it by setting the following hypothesis: H_0: HP-BA$_{\beta_1}$ - HP-BA$_{\beta_2} = \mu$, where $\beta_1 \neq \beta_2$, to express the superiority of the version HP-BA$_{\beta_1}$ according to HP-BA$_{\beta_2}$ and, the hypothesis $\overline{H}_0$: the version of HP-BA$_{\beta_2}$ performs better than the version HP-BA$_{\beta_1}$ to underline the rejection of the hypothesis H_0. Because we study a maximization problem, we deduce that the greatest the provided bound and the greater the achieved number of better bounds, the better the corresponding version of the method (HP-BA).

	μ_1	μ_2	μ_3	μ_4	μ_5
p-value (Sign test)	0.965	0.855	0.637	0.996	0.016
N^+	2	3	4	1	6
N^-	6	5	4	7	0
$N^=$	2	2	2	2	4
p-value (Wilcoxon)	0.869	0.663	0,800	0.987	0.014

Table 3.4: p-values for both *sign* and *Wilcoxon rank* tests on some instances with the significance level $\alpha = 0.05$.

Table 3.4 reports, by varying β_1 and β_2 ($\beta_1 \neq \beta_2$), the study on some instances extracted from the literature, where both *sign test* and *sign rank test* (their corresponding average bounds are those illustrated in Tables 3.3) are considered. The results reported in the table obtained according to the following study: (i) making a comparative study between two versions according to the values assigned to the parameter β and (ii) returning the best version of the method (i.e., the β inducing a better performance) and continuing the comparison with the untested version. The head of the table (line 0) expresses what follows: $\mu_1 = \mathrm{HP-BA}_{5\%}$ vs $\mathrm{HP-BA}_{10\%}$, $\mu_2 = \mathrm{HP-BA}_{10\%}$ vs $\mathrm{HP-BA}_{15\%}$, $\mu_3 = \mathrm{HP-BA}_{15\%}$ vs $\mathrm{HP-BA}_{30\%}$, $\mu_4 = \mathrm{HP-BA}_{30\%}$ vs $\mathrm{HP-BA}_{50\%}$ and $\mu_5 = \mathrm{HP-BA}_{50\%}$ vs $\mathrm{HP-BA}_{60\%}$. Columns from 2 to 6 show the statistical results of the proposed method HP-BA when varying the parameter β, where the first line of the table (line 1) reports the p-value corresponding to the sign test, the second line (line 2) displays the number of times that the first version of HP-BA with its corresponding β dominates the second version with its corresponding β, the third line (line 3) of the table tallies the number of times that the first version is dominated by the second one, the fourth line (line 4) displays the number of times that both versions are able to match the same average bounds while the last line (line 5) reports the p-value related to the rank sign test. From Table 3.4, we observe what follows:

- For μ_1 the p-value related to the sign test (resp. Wilcoxon rank-test) is greatest to the significance level $\alpha = 0.05$, indicating that HP-BA$_{10\%}$ performs better than HP-BA$_{5\%}$ (rejecting the hypothesis H_0).
- For μ_2 the p-value related to the sign test (resp. Wilcoxon rank-test) is also greatest to the significance level $\alpha = 0.05$, indicating that HP-BA$_{10\%}$ performs better than HP-BA$_{15\%}$ (rejecting the hypothesis H_0). In this case, HP-BA$_{15\%}$ outperforms HP-BA$_{5\%}$.
- Increasing the value of β improves the quality of the achieved average bounds. In this case, when comparing the three next versions of HP-BA (i.e., $\mathrm{HP}-\mathrm{BA}_{30\%}$, $\mathrm{HP}-\mathrm{BA}_{50\%}$ and $\mathrm{HP}-\mathrm{BA}_{60\%}$), one can observe that $\mathrm{HP}-\mathrm{BA}_{50\%}$ performs better than $\mathrm{HP}-\mathrm{BA}_{30\%}$ according to Wilcoxon's p-value related to μ_4 (that is equal to 0.987 for $\mathrm{HP}-\mathrm{BA}_{30\%}$) and $\mathrm{HP}-\mathrm{BA}_{50\%}$ outperforms $\mathrm{HP}-\mathrm{BA}_{60\%}$ regarding μ_4 (the Wilcoxon's p-value is closest to 0.014).
- The number of average bounds matched by HP-BA$_{30\%}$, when compared its result to the version HP-BA$_{30\%}$ and HP-BA$_{60\%}$, varies from 6 to 7 for N^+ representing $(1-\mu_4)$ and μ_5. It means that $\mathrm{HP}-\mathrm{BA}_{50\%}$ represents the best version of the algorithm.

Hence, because we are looking for the version, which achieve solutions with high quality, with greatest objective values, and with more best average lower bounds, we then decide to fix $\beta = 50$ for the rest of the computational results.

3.4.2 Performance of the first version of HP-BA vs better available methods

In order to evaluate the behavior of the first version of HP-BA, we compared its provided results to those achieved by the more recent method designed in Atta et al. (2022) (noted MMA) and the state-of-the-art Cplex solver (noted Cplex). Tables 3.5 and 3.6 report the solution values achieved by Cplex, MMA and HP-BA on the medium-sized instances (Table 3.5) and the large-sized instances (Table 3.6), where their characteristics are detailed in Table 3.1. For each table, columns 1 and 2 report the instance's information, column 3 displays the best percentage coverage (lower bound) achieved by the Cplex solver, column 4 displays the best percentage coverage provided by MMA (extracted from Atta et al. (2022)), while the rest of the columns show the best percentage coverage (Max, column 5) reached by HP-BA and its average percentage coverage (Av, column 6) reached over the ten trials considered.

First, in what follows, we comment on the results of Tables 3.5 and 3.6:

- Medium-sized instances:
    1. Column 5 of Table 3.5 shows the competitiveness of HP-BA when comparing its achieved results to those reached by the Cplex solver. Indeed, its global average percentage coverage is equal to 83.47 while that of Cplex is equal to 83.05. Moreover, HP-BA matches several lower bounds of the literature and dominates the state-of-the-art solver by achieving 18 better bounds, representing a percentage of 62.07% of the best provide bounds.
    2. HP-BA's global average percentage coverage remains also better than that achieved the best method of the literature MMA, where its provided global average bound is

Table 3.5: Performance of HP-BA versus both Cplex and the best method of the literature: medium-sized instances

m	k	% Coverage Cplex	MMA	HP-BA Max	HP-BA Av.
324	1	**37.03**	**37.03**	**37.03**	37.01
	2	65.14	65.14	**65.14**	65.14
	3	86.24	86.24	**86.25**	85.96
	4	92.58	**93.28**	93.10	91.96
	5	98.24	98.82	**99.07**	96.97
	6	99.72	**99.78**	99.54	99.43
	7	99.99	**100.00**	99.96	99.92
402	1	**28.15**	**28.15**	**28.15**	28.15
	2	56.10	56.10	56.10	56.03
	3	77.25	77.25	77.25	75.48
	4	91.59	91.78	**91.81**	90.90
	5	96.74	96.83	**96.98**	96.15
	6	**99.39**	99.32	**99.39**	99.39
	7	99.47	**99.93**	**99.93**	99.90
	8	99.86	**100.00**	99.73	99.72
	9	99.71	**100.00**	99.77	99.69
500	1	**22.83**	**22.83**	**22.83**	22.83
	2	**45.67**	**45.67**	**45.67**	45.63
	3	66.44	66.44	66.44	66.40
	4	80.01	79.94	**80.19**	79.46
	5	87.11	**88.58**	**88.58**	88.58
	6	91.68	92.83	**93.04**	92.87
	7	94.90	95.80	**96.06**	96.05
	8	97.00	98.24	**98.58**	98.57
	9	97.54	**99.26**	99.03	99.03
	10	99.42	**99.72**	**99.72**	99.72
	11	98.69	**99.85**	**99.85**	99.55
	12	99.97	**99.98**	**99.98**	99.98
	13	99.98	**100.00**	**100.00**	99.82
Av.		83.05	83.41	**83.47**	83.12

equal to 83.41, which remains slightly smaller than that achieved by HP-BA (i.e., 83.47). In this case, HP-BA reaches 12 better lower bounds, matches 10 bounds (achieved by MMA) and fails in 6 occasions.

3. HP-BA vs both Cplex and MMA: overall instances tested, HP-BA is able to discover 13 new bounds (column 5, in bold-space), it achieves the same bounds in 11 occasions while it fails only in 6 occasions.

- Large-scale instances:

1. HP-BA remains competitive when comparing its achieved results to those reached by both Cplex solver and the best available values in the literature (MMA), especially for large-scale instances. Indeed, for overall tested instances (of this group), HP-BA's global average percentage coverage is equal to 77.08 while that of MMA is equal to 76.87, as underlined in the last line of Table 3.6. Further, when comparing the results reached by HP-BA to those provided by Cplex and MMA, one can observe that HP-BA is able to reach 26 (resp. 28) better lower bounds, matches 18 (resp. 24) bounds and fails in 11 (resp. 3) occasions.

Table 3.6: Performance of HP-BA versus both Cplex and the best method of the literature: large-scale instances

		% Coverage			
				HP-BA	
m	k	Cplex	MMA	Max	Av.
708	1	18.60	18.60	18.60	18.58
	2	37.20	37.20	37.20	37.19
	3	55.80	55.80	55.80	55.80
	4	70.86	70.70	70.70	68.51
	5	80.89	81.33	81.48	81.27
	6	85.28	86.05	86.51	86.51
	7	n/a	88.97	89.11	89.11
	8	n/a	92.13	91.82	91.82
	9	n/a	94.48	94.62	94.62
	10	n/a	97.20	97.66	97.66
	11	n/a	97.83	98.39	98.39
	12	n/a	98.35	98.39	98.39
	13	n/a	99.34	99.44	99.44
	14	n/a	99.59	99.51	99.51
	15	n/a	99.86	99.29	99.29
818	1	15.43	15.43	15.43	15.43
	2	30.86	30.86	30.86	30.86
	3	46.28	46.28	46.28	46.28
	4	59.97	59.97	59.97	59.97
	5	72.42	72.00	72.00	72.00
	6	81.04	81.00	81.00	81.00
	7	84.30	85.61	85.61	85.40
	8	n/a	89.05	89.35	89.35
	9	n/a	92.15	92.26	92.08
	10	n/a	93.35	93.82	93.78
	11	n/a	94.82	95.46	95.46
	12	n/a	95.98	96.40	96.40
	13	n/a	97.11	97.11	97.11
	14	n/a	98.80	98.82	98.03
	15	n/a	99.57	99.57	99.57
700	1	17.01	17.01	17.01	17.01
	2	34.02	34.02	34.02	34.02
	3	51.03	51.03	51.03	51.01
	4	68.04	68.04	68.04	67.99
	5	82.38	83.36	83.80	83.01
	6	85.88	87.44	89.09	87.40
	7	87.06	89.54	90.40	90.30
	8	88.64	91.69	92.73	92.31
	9	89.30	93.21	93.55	92.86
	10	90.23	94.07	94.28	94.07
	11	91.03	94.60	94.61	94.12
	12	91.67	95.36	95.51	94.97
900	1	13.18	13.18	13.18	13.18
	2	26.35	26.35	26.35	26.35
	3	39.53	39.53	39.53	39.53
	4	52.70	52.70	52.70	52.68
	5	65.88	65.88	65.88	65.88
	6	79.05	79.05	79.05	79.05
	7	92.22	92.22	92.22	92.22
	8	97.18	97.54	98.29	98.29
	9	97.43	98.29	98.43	98.42
	10	97.77	98.24	98.94	98.93
	11	98.21	98.55	99.26	99.25
	12	98.28	98.84	99.52	99.49
	13	98.53	98.86	99.33	99.31
	Av.	67.67	68.41	**68.64**	68.43
	Av. Glo.		76.87	**77.08**	76.92

2. Cplex vs other methods: because the Cplex solver fails to solve some large-scale instances (the symbol n/a, column 3), we then reported the global average bounds only on the instances for which the solver was able to solve them within the runtime limit. At the end of the table (Line Av.) reports the global average bounds reached by the three tested methods. For this study, we can also observe that HP-BA performs better than both Cplex and MMA.

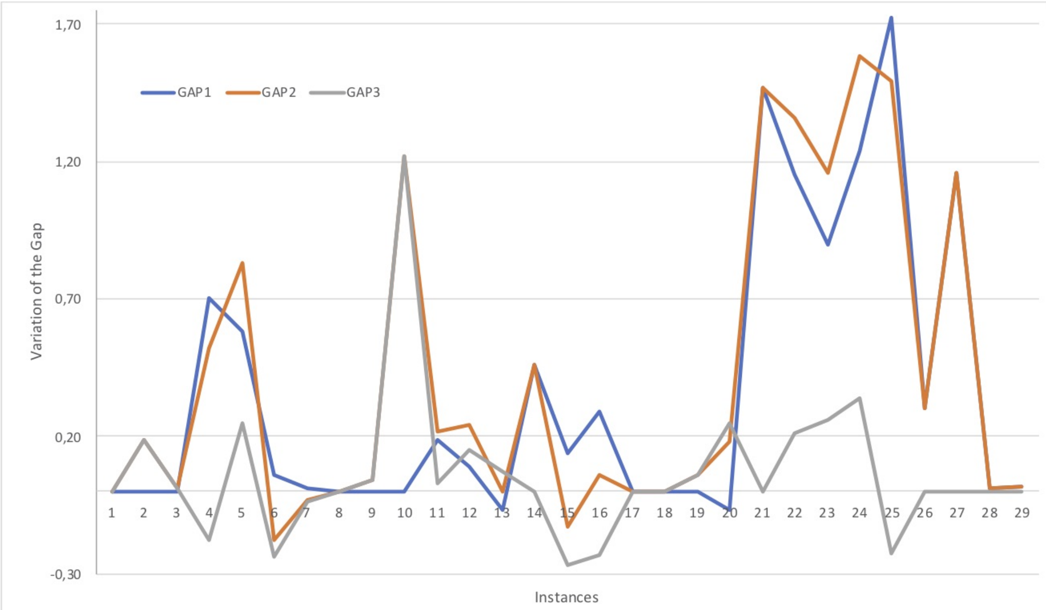

Figure 3.4: Variation of the different Gaps between the three tested methods (Cplex, MMA and HP-BA) on medium-sized instances.

Figure 3.4 shows the variation of the different gaps on the first group of instances (medium-sized instances), where Gap1 = $\%\mathrm{Cov}_{\mathrm{MMA}}$ - $\%\mathrm{Cov}_{\mathrm{Cplex}}$, Gap2 = $\%\mathrm{Cov}_{\mathrm{HP-BA}}$ - $\%\mathrm{Cov}_{\mathrm{Cplex}}$ and, Gap3 = $\%\mathrm{Cov}_{\mathrm{HP-BA}}$ - $\%\mathrm{Cov}_{\mathrm{MMA}}$, respectively. These gaps are computed according to best lower bounds achieved by the three methods and detailed in Table 3.5.

By considering the third gap (Gap3) expressing the difference between the lower bounds provided both MMA et HP-BA on the instances unsolved by the Cplex solver, Figure 3.4 illustrates the variation of that gap, which are obtained according to best lower bounds achieved by both MMA and HP-BA on some of the instances (detailed in Table 3.6).

Finally, Figure 3.5 illustrates the variation of the three gaps (Gap1, Gap2 and Gap3) on the second group of instances (large-scale instances), where these gaps are calculated according to the overall best bounds provides by the Cplex, MMA and HP-BA on overall instances of Table 3.6.

Second, in order to complete the experimental study, we also made a statistical analysis on the results provided by the three methods. In Table 3.7, columns from 2 to 4 report the statistical results of HP-BA when compared to the Cplex solver and MMA: line 1 (resp. line 4) tallies the p-value corresponding to the sign test (resp. rank sign test), line 2 displays the number of times that the first algorithm tested dominates HP-BA while line 3 reports the number of times that the first algorithm is dominated by HP-BA. In this case, we set μ_1 =Cplex vs HP-BA, μ_2 =Cplex vs MMA, and μ_3 =MMA vs HP-BA. Of course, we distinguished both cases: (i) the case where the method is able to solve (approximately) all instances (MMA vs HP-BA) and the (ii) the case with the solved instances (Cplex vs MMA, and Cplex vs HP-BA).

From Table 3.7, we observe what follows:

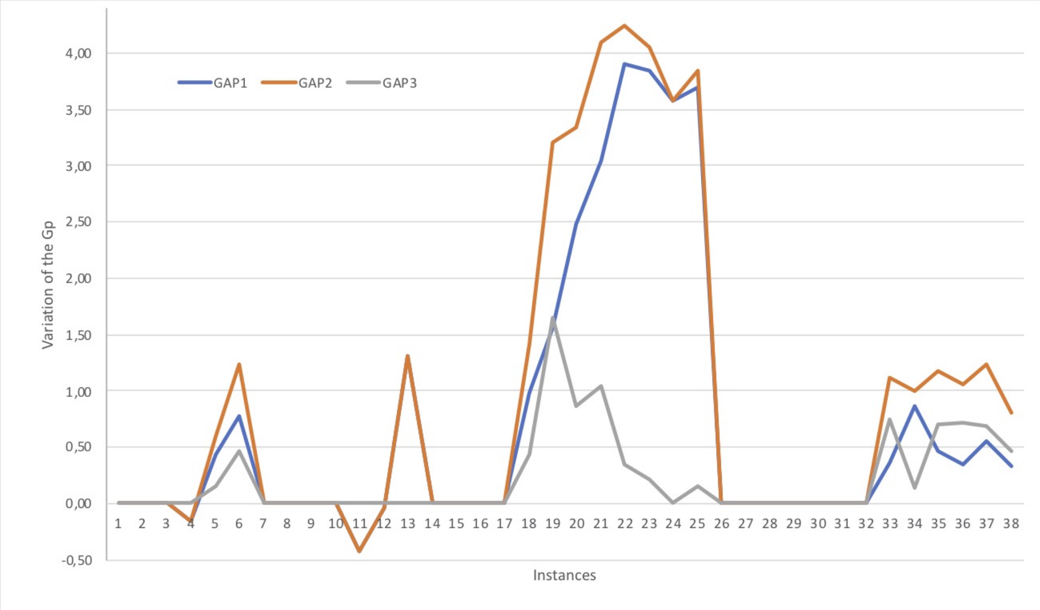

Figure 3.5: Variation of the different Gaps between the three tested methods (Cplex, MMA and HP-BA) on some large-scale instances.

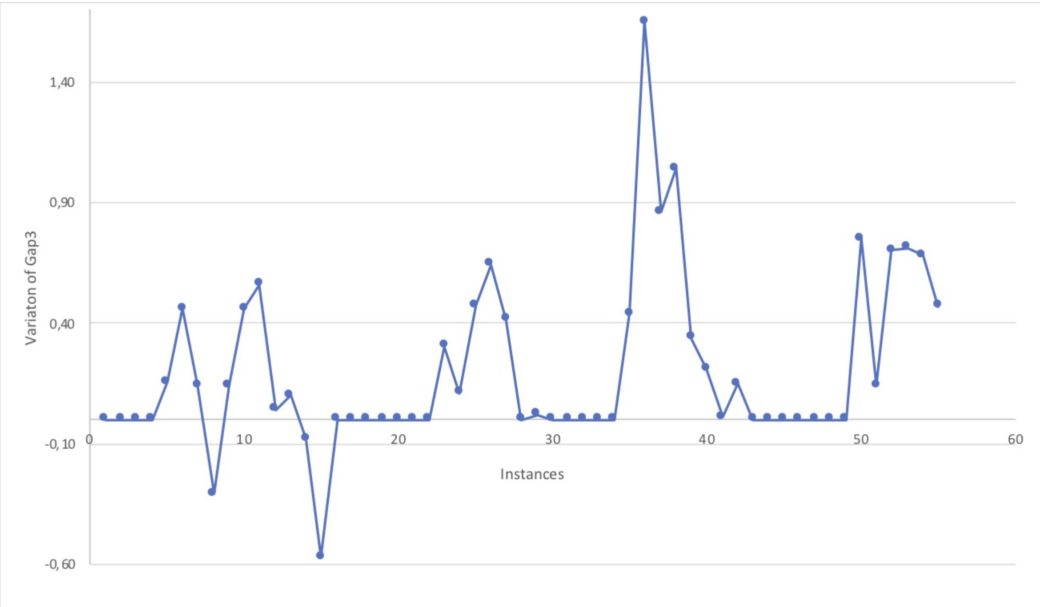

Figure 3.6: Variation of the Gap representing the difference between both MMA and HP-BA on overall large-scale instances.

- p-values related to μ_1, μ_2 and μ_3 are greater than the significance level $\alpha = 0.05$ for both sign and Wilcoxon test. These vales indicate that HP-BA performs better than both Cplex and MMA, by rejecting the hypothesis H_0 in all the experiments.

- The number of occasions that HP-BA matches the best average bound is equal to 32 and 34 (line 3, columns 3 and 4) respectively. It also matches the same average bounds in 29 and 41 occasions, respectively, when compared to Cplex and MMA.

	μ_1	μ_2	μ_3
p-value (Sign test)	1.000	1.000	1.000
N^+	5	6	9
N^-	27	32	34
$N^=$	35	29	41
p-value (Wilcoxon)	1.000	1.000	1.000

Table 3.7: Behavior of HP-BA versus Cplex and MMA: p-values for both *sign* and *Wilcoxon rank* tests on both groups of instances, with the significance level $alpha = 0.05$.

Hence, the statistical analysis confirms the competitiveness of HP-BA on overall benchmark instances.

Effect of the upper bound on the population-based method: HP-BA$_2$

As mentioned in Section 3.3.2.1, the study focuses on the design of two versions of HP-BA. The first version studied and analyzed in the first part of the experimentation showed its competitiveness when its results were compared to those obtained by the best methods available in the literature. We study here the behavior of the second version (noted HP-BA$_2$) of the method, which consists in performing a random generation of the starting reference set around an upper bound. From a first solution built according to a fractional variable related to the facilities to be opened (k facilities), the algorithm tries to delimit the positions of the wolves around this bound, in order to avoid the divergence of the wolves.

Table 3.8: Behavior of the second version of HP-BA (HP-BA$_2$) compared to the first version.

m	k	UB	HP-BA	HP-BA$_2$	A(HP-BA)	A(HP-BA$_2$)
324	4	100	93.1	93.28	0.9310	0.9328
500	2	45.67	45.66	45.67	0.9998	1
700	8	100	92.73	93.1918	0.9273	0.9319
	9	100	93.55	93.92	0.9355	0.9392
	10	100	94.28	94.8845	0.9428	0.9488
	11	100	94.61	95.3122	0.9461	0.9531
	12	100	95.51	95.97	0.9551	0.9597
900	2	26.3521	26.35	26.351	0.9999	1
	9	100	98.431	98.75	0.9843	0.9875
	10	100	98.943	99.15	0.9894	0.9915
	11	100	99.261	99.75	0.9926	0.9975
Average			84.7659	85.1118	0.9640	0.9675

Table 3.8 reports the results provided by the second version of HP-BA (noted HP-BA$_2$) and those achieved by the first version HP-BA. In this case, only improved lower bounds achieved by HP-BA$_2$ are reported (for all the rest of the instances both HP-BA and HP-BA$_2$ match the same lower bounds). Columns 1 and 2 report the instance information. Column 3 displays the optimal solution of $\mathrm{RP}_{\mathrm{FCMCLP}}$ (which is an upper bound for FCMCLP). Columns 4 and 5 tally the best lower bounds provided by both HP-BA and HP-BA$_2$. Finally, columns 6 and 7 show the experimental approximation ratios realized by HP-BA and HP-BA$_2$, respectively. Of course, because the optimal values are unknown (except for some instances, like 500.2 and 900.2, for which the methods match the upper bound), we then, compute each approximation ratio as

follows: $\frac{z(\text{HP}-\text{BA})}{UB}$ for HP-BA and $\frac{z(\text{HP}-\text{BA}_2)}{UB}$ for HP-BA$_2$. In what follows, we comment on the results of Table 3.8:

1. HP-BA$_2$ is able to provide 10 new lower bounds, when compared its results to those achieved by HP-BA (and the best bounds available in the literature). In this case, two optimal solutions are reached by HP-BA$_2$, because it matches both upper bounds (instances 500.2 and 900.2).

2. From the last two columns, one can observe that both HP-BA and HP-BA$_2$ achieve an approximation experimental ratio varying from 0.9273 (instance 700.8) to 0.9999 (instance 900.2) with the first version of the algorithm, while it varies from 0.9319 (instance 700.8) to 1 (instances 500.2 and 900.2) with the second version.

Finally, Figure 3.7 illustrates the variation of the experimental approximation ratios of both HP-BA and HP-BA$_2$ on instances for which HP-BA$_2$ dominates the first version of HP-BA.

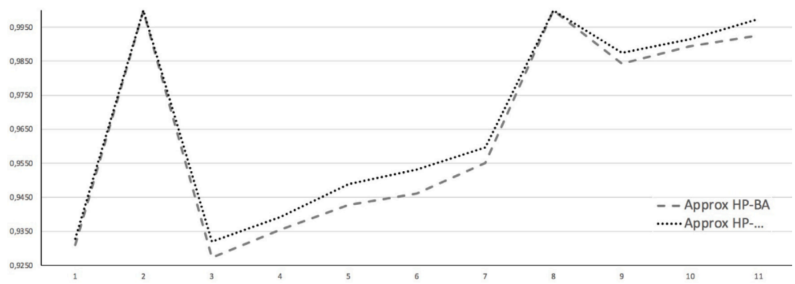

Figure 3.7: Variation of the experimental approximation ration of both HP-BA and HP-BA$_2$ on some instances of the literature.

3.5 Conclusion

In this Chapter, the fuzzy capacitated maximal covering location problem was solved with a hybrid population-based algorithm. The method hybridized a discrete grey wolf optimizer with local operators; it combines three main futures. First, a diversified starting population was built by tailoring a random constructive procedure. Second, an exploration search was added in order to augment the quality of the positions of wolves, which characterizing better solutions for the problem; in this case, a special tabu list with a hashing function were also incorporated for avoiding cycling on some visited solutions. Third and last, the drop and rebuild operator was added as an exploration strategy for searching un-visited sub-spaces around a series of preys' positions related to the three best wolves' positions. Finally, the performance of the hybrid population-based algorithm was evaluated on a set of benchmark instances, where its achieved results were compared to those provided by the state-of-the-art Cplex solver and to the best results extracted from a recent paper in the literature. Several new upper bounds have been discovered.

Chapter 4

A Population-Based Algorithm for Solving the Fuzzy Capacitated Maximal Covering Location Problem

4.1 Introduction

The Fuzzy Capacitated Maximal Covering Location Problem (FCMCLP) can be viewed as an extended version of MCLP, where its goal is to determine the best location of a subset of given facilities that covers a maximum number of customers such that both the distance between the customers and the installed facilities, and the coverage area of each facility are fuzzy.

In this chapter, we tackle FCMCLP with an evolutionary algorithm, which is based on a hybrid grey wolf optimizer Aïder et al. (2022). An instance of FCMCLP is characterized by a set P of m points in a plane and a set of k, $1 \leq k \leq m$ facilities to be installed at a chosen subset of points. Each customer $p_i \in P, i = 1, \ldots, m$ is represented by a non-negative demand w_i while each located facility is represented by a supply capacity s_j, $1 \leq j \leq m$. A facility can be positioned within its coverage area center c_j, which is considered as two concentric circles, with R radius and $R + f_r$ fuzzy radius, respectively. The centers of the aforementioned circles are restricted to the locations of customers themselves (Lorena and Pereira, 2002) while the coverage degree of a facility c_j positioned at a point p_i, based on the definition of Drakulic et al. (2016), is given as follows:

$$Cov(i,j) = \begin{cases} 1 & \text{if } d(p_i, c_j) \leq R \\ \frac{1}{2}(\mu(x_1) + \mu(x_2)) & R < d(p_i, c_j) \leq R + f_r \\ 0 & d(p_i, c_j) > R + f_r, \end{cases} \tag{4.1}$$

Where, on the one hand, $d(p_i, c_j)$ denotes the Euclidean distance between the positioned facility c_j and the customer p_i, and on the other hand, $d(p_i, c_j) \pm f_d$ denotes its fuzzy distance Drakulic et al. (2016) (Figure 3.1 illustrates both values $\mu(x_1)$ and $\mu(x_2)$ used by the formula (4.1)).

The rest of the chapter is organized as follows: A brief mathematical formulation of the FCMCLP is presented in Section 4.1. In Section 4.2, we first recall the basic principle of the grey wolf optimizer, and next, we present the evolutionary algorithm designed for approximately solving FCMCLP. In the same section, an enhanced version of the evolutionary algorithm is presented. Finally, in Section 4.3, the behavior of the designed algorithm is tested and analyzed for instances taken from the literature.

A formal description of FCMCLP

We give a formal description of FCMCLP, already considered in the literature (See 2). Let x_{ij} be a decision variable related to linking the customer's position p_i and the facility c_j such that

$$x_{ij} = \begin{cases} 1 & \text{if customer } i \text{ is served by facility } j \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

and y_j be a decision variable related to a facility j such that

$$y_j = \begin{cases} 1 & \text{if a facility is opened at the location j} \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

A simple FCMCLP's formal description is as follows:

$$\max \qquad z = \sum_{i=1}^{m} \sum_{j=1}^{m} (w_i \times cov_{ij}) x_{ij} \tag{4.4}$$

$$\text{s.t.} \qquad \sum_{j=1}^{m} y_j = k \tag{4.5}$$

$$\sum_{j=1}^{m} x_{ij} \leq 1, \forall i \in I \tag{4.6}$$

$$\sum_{i=1}^{m} (w_i \times cov_{ij}) x_{ij} \leq s_j y_j, \forall j \in J \tag{4.7}$$

$$x_{ij} \in \{0,1\}, y_j \in \{0,1\},\ i \in I, j \in J, \tag{4.8}$$

where $I = J = \{1, ..., m\}$ and s_j is the supply (capacity) value of the j-th facility.
For more details, readers may refer to section 3.3.1 .

4.2 An Evolutionary Algorithm for FCMCLP

In this section, we first describe the basic principle of the grey wolf optimizer, and then we discuss how such a method can be adapted (and enhanced) for tackling the FCMCLP.

4.2.1 A basic grey wolf optimizer

As it has already been mentioned in 3.3.2, the Grey Wolf Optimizer (GWO) is a new effective meta-heuristic inspired from the grey wolves in nature. It tries to mimic the leadership hierarchy and hunting mechanism of grey wolves and to use it as a search process for generating a series of positions related to feasible solutions to a given problem Seyedali Mirjalili and Lewis (2014). In addition to the social hierarchy of wolves, group hunting is another interesting social behavior of grey wolves. According to Muro et al. (2011), the main phases of grey wolf hunting are represented by tracking, encircling, and attacking the prey.
The main phases of hunting are given as follows (in order to encircle the prey, the wolves use a precise direction as described below):

$$\overrightarrow{D} = |\overrightarrow{C} \cdot \overrightarrow{X_p}(t) - \overrightarrow{X}(t)| \tag{4.9}$$

At each iteration, the position of each wolf will be updated:

$$\overrightarrow{X}(t+1) = \overrightarrow{X_p}(t) - \overrightarrow{A} \cdot \overrightarrow{D}, \tag{4.10}$$

Where t denotes the current iteration, $\overrightarrow{X_p}$ is the position vector of the prey, $\overrightarrow{X}$ represents the position vector of the wolf, while $\overrightarrow{A}$ and $\overrightarrow{C}$ are two coefficient vectors defined as follows:

$$\overrightarrow{A} = 2\overrightarrow{a} \cdot \overrightarrow{r_1} - \overrightarrow{a} \quad \text{and} \quad \overrightarrow{C} = 2\overrightarrow{r_2} \tag{4.11}$$

Often, $\overrightarrow{a}$ (used in the left-hand of Eqs (4.11)) is a vector initialized to 2 and decreases till matching the value 0 throughout the iterations, while $\overrightarrow{r_1}$ (on the left-hand of Eqs (4.11)) and $\overrightarrow{r_2}$ (on the right-hand) denote two vectors whose components are randomly generated in the continuous interval $[0, 1]$. Next, at each iteration of the search process, the positions of α, β and δ wolves are updated according to Eq (4.10). Finally, the position of the prey is updated (at iteration $t + 1$) as follows:

$$\overrightarrow{X_p}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3}, \tag{4.12}$$

where its position is scattered by combining the new positions of the previous three best wolfs of the group.

4.2.2 An evolutionary algorithm for FCMCLP

GWO works with a group of wolfs, which can be viewed as a population-based method. Hence, GWO may first start with a randomly generated population $\mathcal{P}$ of wolfs (solutions), using a generator trying to cover the search space of the problem concerned for diversification purposes (Section 4.2.2.1). Second, at each step, regarding the position of the prey, the positions of the wolfs should be computed and the new prey's position is also calculated by combining the three best positions (according to their finesses) in the population (Section 4.2.2.2). Third, the new population is ranked in decreasing order of their finesses (a maximization problem). Fourth and last, such a process is iterated till matching the stopping criteria.

In what follows, we try to mimic such a process for achieving a near-optimal solution for FCMCLP.

4.2.2.1 A starting population

Because GWO is a stochastic method, we then randomly generate a set $\mathcal{P}$ of feasible solutions, where each solution is assigned to a wolf's position. The starting population $\mathcal{P}$ is generated according to the following steps, where $\overline{S}^{(i)}$ represents the i-th wolf (solution):

1. Set $\mathcal{P} = \emptyset$ (the population size $\rho = |\mathcal{P}| > 0$).
2. Set $i = 0$ and $nb_{\text{Iter}} = 0$ (the maximun iterations was fixed to Max-nb_{Iter}

   Do Increment i and set $\overline{S}^{(i)} = \emptyset$.

   (a) **Do** Set $\ell =$ `rand`$(1, n)$; Check the feasibility when opening ℓ for $\overline{S}^{(i)}$, close (remove) it (from $\overline{S}^{(i)}$) otherwise.
   Until (providing a complete solution).

   (b) If $\overline{S}^{(i)} \notin \mathcal{P}$, then set $\mathcal{P} = \mathcal{P} \cup \overline{S}^{(i)}$ (decrement i otherwise).
   Increment nb_{Iter}.

   Until ($i = \rho$ or nb_{Iter} =Max-nb_{Iter}).

4.2.2.2 Generating new positions

Determining a new prey's position is equivalent to combine the new positions of the three best solutions. Often, Eq (4.12) induces a fractional solution whenever at least one component, at the same rank of each solution, was fixed to one. Also, evaluating the new wolf's position depends on the position of the prey, where applying Eq. (4.10) may induce fractional positions. Herein, we make a cooperation between a series of fractional positions of the prey and the integral positions of the wolves. Each new integral position i may be provided as follows:

$$\overline{S}^{(i)} = \left(\left\lceil \overline{S}_1^{(i)} \right\rceil, \ldots, \left\lceil \overline{S}_n^{(i)} \right\rceil \right), \tag{4.13}$$

where $\lceil x \rceil$ denotes the greatest integer value related to x.

One can observe that each component $\overline{S}^{(i)} = \lceil x \rceil$ may provide a value out of the discrete interval $\{1, \ldots, n\}$, where that value reflects the assignment of a facility to a customer. Hence, in order to work with feasibility, we propose the following greedy operator to correct the solution:

1. Let $\overline{S}^{(i)}$ be the current integral position provided by applying Eq (4.13). Suppose that all component of $\overline{S}^{(i)}$ are unfixed (free).
2. Reorder the customers as follows: $\frac{w_1^{(i)}}{\delta_1^{(i)}} \geq \ldots \geq \frac{w_n^{(i)}}{\delta_n^{(i)}}$, where $\delta_k^{(i)}$ (resp. w_k) denotes the degree (resp. weight) of the customer k of the i-th configuration, $k = 1, \ldots, n$. Set $\ell = \arg \max_{1 \leq k \leq n} \left\{ \frac{w_k^{(i)}}{\delta_k^{(i)}} \right\}$.
3. Check feasibility by fixing the facility ℓ, to close it otherwise.
4. Repeat steps (2) and (3) till providing a complete solution (fixing −closed or open−all possible facilities).

4.2.3 Intensification Search

Searching for an improved solution (integral position) is equivalent to fix some stations among the n possible stations. Making some moves between fixed stations is equivalent to fix some of them and to close the rest of the facilities.

4.2.3.1 A 2-opt Operator

It can be viewed as a simple local search, which is even based upon basic local modifications of the solution at hand. Often, the operator sequentially makes a series of swaps as long as the quality of the induced solution is enhanced. Whenever the solution stagnates where its objective value remains quasi-unchanged, then the operator converges to a local optimum. Herein, we propose to swap between a current opened facility ℓ of the i-th solution $\overline{S}^{(i)}$ with a chosen facility satisfying the following criteria:

$$\ell = \arg \max_{1 \leq k \leq n; k = \text{unfixed}} \Big\{ Cov(k, r),\ r \in \{1, \ldots, n\}, \\ k \neq r,\ r = \text{fixed} \Big\}. \tag{4.14}$$

Of course, for each new opened facility ℓ, the procedure checks the feasibility related to the induced solution.

4.2.3.2 Avoid cycling

Often, the 2-opt operator tries to build a series of solutions around other solutions. A subset of solutions may be viewed as a neighborhood, according to a given solution. Because a new solution built can be provided by swapping two facilities, one can observe that repeating the same process may lead toward the same local optimum and so, the operator may provide a premature convergence towards local optimum. Herein, we add to the 2-opt operator a tabu list in order to mimic the well-known tabu search Glover (1989), where the tabu list contains a list of temporarily reverse swaps that avoids to return to the solutions already visited.

4.2.4 Diversification Search

The intensification goal is to provide a series of neighborhoods, which might contain solutions with better quality. A diversification strategy tries to drive the search process through other un-visited sub-spaces. We simulate simulate such a diversified operator by adding the Drop and Rebuild Operator (DRO) already used in Hifi (2002), where it aims at removing a part of assigned facilities from the current solution. In this case, a first partial (incomplete) solution is built; that is completed by using a part of the procedure described in Section 4.2.2.2. It can be summarized by the following steps:
1) Let $\overline{S}^{(i)}$ be the current solution.
2) Drop β% of the fixed facilities belonging to $\overline{S}^{(i)}$.
3) Complete the partial solution by adding the facilities satisfying the best weight per degree, i.e., facility ℓ such that

$$\ell = \arg \max_{1 \le k \le n,\ k = \text{non-fixed}} \left\{ \frac{w_k^{(i)}}{\delta_k^{(i)}} \right\}.$$

4.3 Computational Results

The proposed Evolutionary Algorithm (EA) is evaluated on five different benchmark instances extracted from Atta et al. (2022), where each set (i) contains between three and four values of k (where $k \in \{1, \ldots, 9\}$); that is the number of facilities to open, (ii) the number of facilities m varies from 324 to 900, $R/f_r/f_d$ (the radius value R and both the fuzzy radius f_r and the distance f_d) vary from 20/10/5 to 600/300/100 and, the supply value S varies from 4500 to 9500. These groups are differentiated with the values assigned to m. The proposed method was coded in c++ and run on an Intel(R) Core(TM) i5-6200U with 2.30 Ghz. Table 3.1 reports overall information related to instances used. (Refer to chapter 3).

Many proposed algorithms have shown different bounds to solve the fuzzy capacitated maximal covering location problem (refer to Atta et al. (2022)). Thus, EAs' results were compared to those of Cplex and the best bounds in the literature. The comparative study was included in Table 4.1.

In order to evaluate the behavior of EA, we compare its provided results to those achieved by the more recent method proposed in Atta et al. (2022) (MMA) and Cplex. Table 4.1 reports the results achieved by both methods, where columns 1 and 2 display the instance information, column 3 (resp. 4) tallies the best percentage coverage (%Cov.) achieved by the Cplex (resp. MMA), and column 5 shows the same bound achieved by EA with its global average value in column 6. Finally, the last three columns display three gaps expressing the difference between the bounds reached by each pair of algorithms: $\text{Gap}_1 = \%\text{Cov}_{\text{MMA}}$ - $\%\text{Cov}_{\text{Cplex}}$, $\text{Gap}_2 = \%\text{Cov}_{\text{EA}}$ - $\%\text{Cov}_{\text{Cplex}}$ and, $\text{Gap}_3 = \%\text{Cov}_{\text{EA}}$ - $\%\text{Cov}_{\text{MMA}}$, respectively. The runtime limit needed for achieving EA's final solution was fixed in one hour, and 2 hours for MMA and Cplex.

Table 4.1: Performance of EA vs both Cplex and MMA

m	k	Cplex %Cov	Best Lit. %Cov	EA: %Cov Av.	EA: %Cov Max	Variation of Gaps Gap_1	Gap_2	Gap_3
324	1	37.03	37.03	37.01	37.03	0	0	0
	2	65.14	65.14	65.24	**65.33**	0	0.19	0.19
	3	86.24	86.24	85.96	**86.25**	0	0.01	0.01
	4	92.58	93.28	91.96	**93.1**	0.7	0.52	-0.18
	5	98.24	98.82	96.97	**99.07**	0.58	0.83	0.25
402	1	28.15	28.15	28.15	28.15	0	0	0
	2	56.10	56.10	56.03	**56.14**	0	0.04	0.04
	3	77.25	77.25	75.48	**78.47**	0	1.22	1.22
	4	91.59	91.78	90.90	**91.81**	0.19	0.22	0.03
	5	96.74	96.83	96.15	**96.98**	0.09	0.24	0.15
500	1	22.83	22.83	22.832	22.83	0	0	0
	2	45.67	45.67	45.63	45.67	0	0	0
	3	66.44	66.44	66.40	**66.50**	0	0.06	0.06
	4	80.01	79.94	79.46	**80.19**	-0.07	0.18	0.25
708	1	18.60	18.60	18.58	18.60	0	0	0
	2	37.20	37.20	37.19	37.20	0	0	0
	3	55.80	55.80	55.63	55.64	0	-0.16	-0.16
	4	70.86	70.70	68.51	68.51	-0.16	-2.35	-2.19
700	1	17.01	17.01	17.00	17.01	0	0	0
	2	34.02	34.02	34.02	34.02	0	0	0
	3	51.03	51.03	51.01	51.03	0	0	0
	4	68.04	68.04	67.99	68.04	0	0	0
	5	82.38	83.36	83.01	**83.80**	0.98	1.42	0.44
	6	85.88	87.44	87.40	**89.09**	1.56	3.21	1.65
818	1	15.43	15.43	15.43	15.43	0	0	0
	2	30.86	30.86	30.86	30.86	0	0	0
	4	59.97	59.97	59.97	59.97	0	0	0
	8	n/a	89.05	**89.35**	**89.35**	n/a	n/a	0.3
	9	n/a	92.15	92.08	**92.26**	n/a	n/a	0.11
900	1	13.18	13.18	13.17	13.18	0	0	0
	2	26.35	26.35	26.34	26.35	0	0	0
	3	39.53	39.53	39.51	39.53	0	0	0
	4	52.70	52.70	52.68	52.70	0	0	0
	Av.	51.60	55.39	54.48	**57.27**	3.79	5.67	1.88

From Table 4.1, one can observe what follows: (**i**) EA is competitive when comparing its achieved results to those reached by the Cplex and the best available bounds in the literature (MMA); (**ii**) EA's average percentage coverage is equal to 57.27 while that of MMA is equal to 55.39: in this case, it achieves an average Gap of 1.88; (**iii**) The global EA's average coverage remains better than that matched by Cplex: Cplex's average coverage is equal to 51.60, which means that EA performs better than Cplex by achieving an average Gap of 5.67; (**iv**) MMA's behavior seems slightly better than Cplex, since it is able to provide 8 better bounds, matches 23 bounds and fails in 2 occasions, meanwhile Cplex fails in solving 2 instances. Further, EA provides 14 better average coverage values, matches 17 bounds according to those achieved by Cplex and fails in 2 occasions; (**v**) EA provides 13 better bounds, matches 14 bounds and fails in 3 occasions. Finally, EA is able to discover 13 new bounds (column 6), achieves the same bounds in 17 occasions while it fails only on 3 occasions.

Figure 4.1 illustrates the variation of Gap 1 (MMA versus Cplex: top-left of the figure), Gap 2 (EA vs Cplex: top-right of figure) and Gap 3 (EA vs MMA: on the bottom of figure).

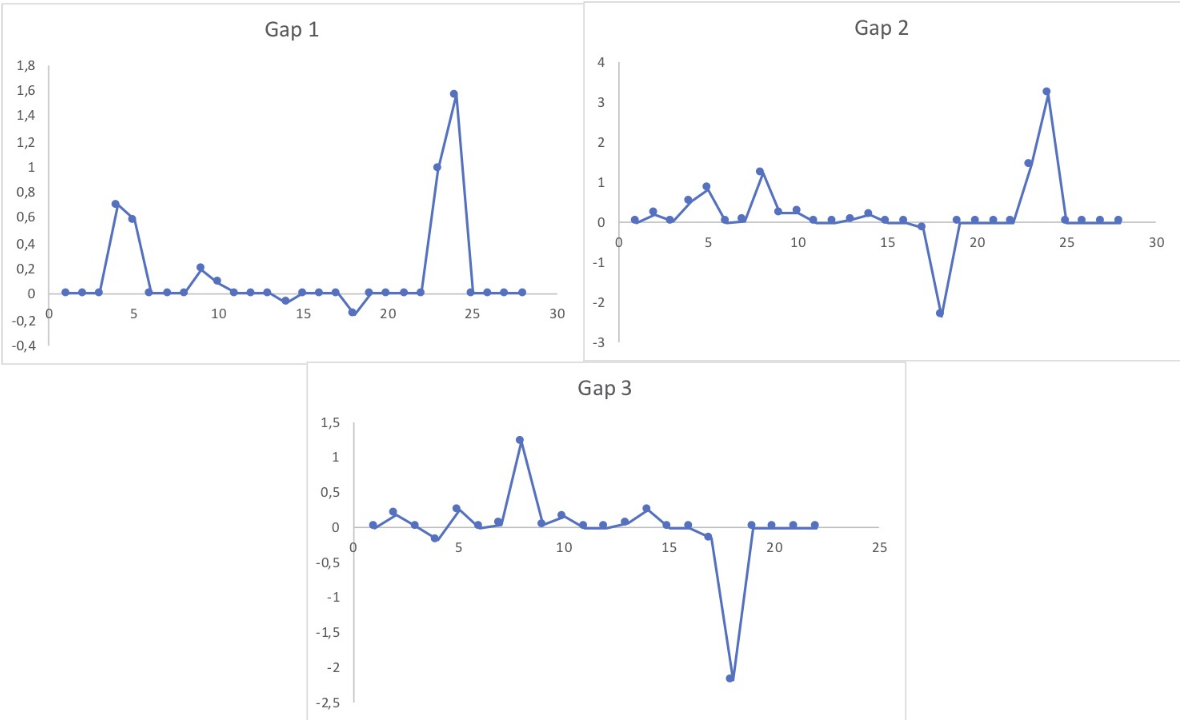

Figure 4.1: Illustration of the variation of the three Gaps achieved by the three methods: Cplex, MMA and EA.

One can observe that EA works with a population size of 10. In fact, in the preliminary study, the behavior of EA was tested on the first 28 instances contained in the six groups. Its behavior was evaluated by varying the size of the population, where initially that population was fixed to the minimum size of 5 and doubled for the next version. For each run, the run-time limit was fixed between ten and thirty minutes (according to the size of the instance). Table 4.2 reports EAs' results, where columns 1 and 2 display the instance information, columns from 3 to 5 show the best upper bounds (percentage coverage, EA_5, EA_{10} and EA_{20}, respectively) achieved by EA according to overall values related to the size of the population, and column 6 to column 8 report EA's run-time when varying the population size. Finally, the last line tallies the average bound of all tested instances for each version of EA and, the average run-time consumed by EA_5, EA_{10} and EA_{20}, respectively.

We then comment on the results of Table 4.2: (**i**) For all considered population sizes, all versions of the EA are able to match several best bounds according to the size of the instance; (**ii**) EA_{10} version (with a population size fixed to 10) reaches a better behavior when comparing its value to those reached by other versions of the algorithm. Indeed, on the one hand, EA_{10} achieves an average value of 57.64, which realizes a Gap (EA_{10}-EA_5) of 0.33 when compared to the average value related to the second best average bounds of EA. On the other hand, EA_{10} seems much faster than the two other versions of EA; in this case, EA needs an average run-time of 6.77 minutes while EA_5 (resp. EA_{20}) consumes 7.69 (resp. 8.54) minutes.

Hence, we can conclude that EA with the population size fixed to 10 has a better behavior for these type of instances.

Table 4.2: Behavior of EA when varying the population size

m	k	%Cov: solution values EA$_5$	EA$_{10}$	EA$_{20}$	Runtime EA$_5$	EA$_{10}$	EA$_{20}$
324	1	37.03	37.03	37.03	1	1	1
	2	63.73	65.33	62.93	2	2	2
	5	98.1	99.07	95.77	4	10	6
402	2	56.04	56.14	56.04	6	6	7
	3	78.47	77.71	77.9	6	8	9
500	1	22.83	22.83	22.83	1	1	1
	4	77.8	80.19	78.92	20	4	20
708	1	18.599	18.6	18.592	2	4	5
	2	37.14	37.2	37.15	6	7	7
700	5	83.8	83.67	83.65	30	25	30
	6	88.46	88.56	89.09	11	8	10
900	2	26.35	26.35	26.35	3	3	3
	4	56.7	56.7	56.7	8	9	10
Average		57.31	**57.64**	57.15	7.69	**6.77**	8.54

4.4 Conclusion

In this chapter, the fuzzy capacitated maximal covering location problem was studied and solved by an evolutionary algorithm. The method hybridized a discrete grey wolf optimizer with local searches; it combines three main futures. First, a starting population was built by tailoring a random constructive procedure. Second, an intensification search was added in order to augment the quality of the positions characterizing better solutions for the problem, where a tabu list was also incorporated for avoiding cycling. Third and last, the drop and rebuild operator was incorporated as diversification strategy, which was added for searching un-visited sub-spaces around a series of preys' positions. Finally, in the experimental part, tests on benchmark instances showed that the proposed algorithm was capable to outperform the Cplex solver and the more recent method of the literature.

Chapter 5

An Extended Rounding Strategy-Based Algorithm for the Fuzzy Capacitated Maximal Covering Location Problem

5.1 Introduction

In this part, we focus on the FCMCLP, where an instance of the problem involves selecting a subset of facilities to be installed at specific points in a plane. Each point represents a customer, and has a non-negative demand associated with it. The facilities have different supply capacities, and can be positioned within their coverage area centers, which are defined as concentric circles.

Readers are referred to section 3.2,to a better comprehension of the presentation of the FCMCLP instance.

5.1.1 Contribution

In recent years, the hybridization of solution procedures has emerged as a promising research area. Combining diverse algorithms with exploration and exploitation operators has shown significant advantages, enabling a balanced approach that generates solutions of both average and high quality. Each study in this field varies in approach, with the aim of designing effective experimental methods that can compete with existing techniques in the literature. In this context, we present a powerful tailored algorithm for solving the FCMCLP (Fuzzy Capacitated Multi-Commodity Location Problem). The designed method starts with an initial solution and expands it by incorporating both exploration and exploitation operators to enhance its effectiveness. The principle of the proposed method is based on the following features:

1. To construct a *starting solution* by applying a *basic rounding procedure* to an initial relaxed problem (cf., Section 5.2.2).

2. The search for an improved lower bound involves exploring the interval $[\alpha_{\min}, \alpha_{\max}]$, where $\alpha_{\min}$ (representing the minimal cardinality constraint for decision variables) and $\alpha_{\max}$ (representing the maximal cardinality constraint) define the range within which decision variables can contribute to a potential optimal solution (cf., Section 5.2.4).

3. Apply a modified descent variable neighborhood search to enhance the quality of a series of solutions. Such a search process employs an incremental procedure, where each established model is built by adding constraints related to both cardinality constraints and those bounding the objective function (cf., Sections 5.2.5 and 5.2.6).

The remainder of the paper is planned as follows. Section 5.2 exposes the proposed rounding strategy-based method for approximately solving the FCMCLP. First, a formal description of FCMCLP is presented in Section 5.2.1. Section 5.2.2 focuses on the discussion of the basic rounding procedure used in this work.
Next, the interval search technique constructed by solving two mixed-integer programming problems is presented in Section 5.2.3. Additionally, an enhanced version of the algorithm is presented in the same section. Finally, Section 5.3 evaluates the performance of the designed algorithm by testing and analyzing its behavior on instances sourced from existing literature. The achieved results are then compared to those provided by recent methods in the literature and the state-of-the-art Cplex solver.

Recently, Aider et al. (2023) introduced a hybrid population-based algorithm to tackle the fuzzy capacitated maximal covering location problem. Two versions of the algorithm based on the grey wolf optimizer have been proposed: (i) a modified grey wolf optimizer with exploration and exploitation strategies, and (ii) a version incorporating local searches and an upper bound to constrain wolf movements. Both versions of the algorithm integrate local searches to significantly enhance solution quality within the current population. Furthermore, the exploration strategy was added, where the drop/rebuild operator is used to facilitate smooth transitions between the current region and unexplored ones. In their experimental analysis, using benchmark instances from the literature, promising results were revealed, with improvements observed in several instances.

5.2 An Extended Rounding Strategy-Based Method

In this section, we first express the mathematical formulation of FCMCLP, already considered in the literature (Atta et al., 2022). Next, we discuss the linear relaxation of the aforementioned model for establishing an upper bound for the hybrid method. Further, we describe the basic principle of the Rounding procedure, including its main steps. Finally, we discuss how such a method can be adapted (and enhanced) for tackling the FCMCLP.

5.2.1 A formal description of FCMCLP

Let x_{ij} be a decision variable related to linking the position p_i of the customer i and the facility c_j such that $x_{ij} = 1$ if customer i is served by facility j, 0 otherwise. Let y_j be a decision variable related to a facility j such that $y_j = 1$ if a facility is opened at the location j, 0 otherwise. Then, a formal description of FCMCLP (noted $\mathrm{P}_{\mathrm{FCMCLP}}$) may be described as follows:

$$\mathrm{P}_{\mathrm{FCMCLP}}: \quad \max \quad z(x) = \sum_{i=1}^{m}\sum_{j=1}^{m}(w_i \times cov_{ij})x_{ij} \tag{5.1}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} y_j = k \tag{5.2}$$

$$\sum_{j=1}^{m} x_{ij} \leq 1, \forall i \in I \tag{5.3}$$

$$\sum_{i=1}^{m}(w_i \times cov_{ij})x_{ij} \leq s_j y_j, \forall j \in J \tag{5.4}$$

$$x_{ij} \in \{0,1\}, y_j \in \{0,1\}, \ i \in I, j \in J, \tag{5.5}$$

For more details, see section 3.3.1.

5.2.2 The Basic Rounding Procedure

The rounding method has been widely employed in various contexts, including binary or mixed-integer problems (c.f., Reihaneh and Ghoniem (2018) and, Tanksale and Jha (2018)). A similar approach to the rounding strategy is observed in Wang (2020), where it is applied to quadratic integer programming problems with linear constraints. In their work, the proposed algorithm integrates the rounding method into a branch-and-bound framework, effectively reducing the solution space. In this paper, we explore an alternative solution approach that draws inspiration from the descent method.

The Basic Rounding Procedure (BRP) utilized in our approach is applied to the mixed linear relaxation of the original problem, denoted as $\mathrm{MIP}_{\mathrm{FCMCLP}}$, which can be described as follows:

$$\mathrm{MIP}_{\mathrm{FCMCLP}}: \quad \max \quad \sum_{i=1}^{m}\sum_{j=1}^{m}(w_i \times cov_{ij})x_{ij} \tag{5.6}$$

$$\text{s.t.} \quad \sum_{j=1}^{m} y_j = k \tag{5.7}$$

$$\sum_{j=1}^{m} x_{ij} \leq 1, \forall i \in I \tag{5.8}$$

$$\sum_{i=1}^{m}(w_i \times cov_{ij})x_{ij} \leq s_j y_j, \forall j \in J \tag{5.9}$$

$$x_{ij} \in [0,1], (i,j) \in I \times J, y_j \in \{0,1\}, \ j \in J, \tag{5.10}$$

Where the decision variables x_{ij} are relaxed. Of course, when all decision variables x_{ij} and y_j are relaxed, we refer to the resulting problem as the relaxed version of the problem $\mathrm{P}_{\mathrm{FCMCLP}}$, denoted as $\mathrm{RP}_{\mathrm{FCMCLP}}$.

The general framework of BRP utilized in this study is customized for tackling the mixed-integer linear relaxation problem, $\mathrm{MIP}_{\mathrm{FCMCLP}}$, and is based on the following steps:

1. Set $P_{\text{reduce}} = \mathrm{MIP}_{\mathrm{FCMCLP}}$.
2. Call the Cplex solver to solve P_{reduce} and extract the solution $(\underline{x}, \underline{y}, \underline{z})$.
3. Assign the value 1 to each $\underline{y}_j$ in $\underline{y}$ if $\underline{y}_j = 1$.

4. Choose the variable $\underline{x}_{it}$ with the largest fractional value among the fractional variables and round it to one.

5. Apply a greedy procedure to the knapsack constraint (5.9) starting from the most recently rounded variable $\underline{x}_{ij}$. This procedure involves fixing all items of $\underline{x}_{ij}$ to 0 while iteratively examining all items in the current knapsack j.

6. Ensure that the fixed values respect the constraint $\sum_{j=1}^{m} x_{ij} \leq 1$.

7. Eliminate all the fixed variables from the current problem, resulting in P_{reduce} as the reduced new problem.

8. Iterate through steps 1 to 7 until all $x_{ij} \in \{0, 1\}$.

The BRP serves as a valuable method to obtain an initial feasible integer solution for P_{FCMCLP}. However, due to the greediness of BRP and the complexity associated with the partitioning constraint of P_{FCMCLP} (represented by Inequality (5.7)), the solutions generated by BRP may reach moderate quality. It is important to note that fixing a decision variable y_j does not directly determine the fixation of its corresponding items, whereas fixing x_{ij} to one enforces the fixation of its associated services. Consequently, BRP may eliminate certain items by fixing their corresponding decision variables to one. As a result, BRP is rarely used as a standalone procedure but rather as an integral component within an efficient computational approach.

5.2.3 Bounding the search space with cardinality constraints

Cardinally constraints can be handled into a model for highlighting the search process used for tackling large-scale and arduous instances. To obtain a tighter bound, we propose to add minimum and maximum cardinally constraints in which the complicated variables are bounded.

On the one hand, starting with the maximum cardinally bound on these part of variables (x_{ij}), assumes that the following constraint is added to P_{FCMCLP}:

$$\sum_{i=1}^{m}\sum_{j=1}^{m} x_{ij} \leq \alpha_{max}, \tag{5.11}$$

where α_{max} is the solution value reached by solving the following model:

$$\begin{aligned}
\text{P}_{\alpha_{max}}: \quad \alpha_{max} = \max \quad & \sum_{i=1}^{m}\sum_{i=1}^{m} x_{ij} \\
\text{s.t.} \quad & (5.2),\ (5.3),\ (5.4), \\
& \sum_{i=1}^{m}\sum_{j=1}^{m} (w_i \times cov_{ij}) x_{ij} \geq LB \\
& x_{ij} \in [0,1]\ \forall (i,j) \in I \times J,\ y_j \in [0,1]\ \forall\, j \in J,
\end{aligned}$$

Where LB denotes the evaluation of the best feasible solution found so far.

On the other hand, starting with the minimum cardinal bound on the variables x_{ij}, one can add the following constraint to P_{FCMCLP}:

$$\alpha_{min} \leq \sum_{i=1}^{m}\sum_{j=1}^{m} x_{ij}, \tag{5.12}$$

where α_{min} is the objective value of the following model:

$$
\begin{aligned}
\mathrm{P}_{\alpha_{min}}: \quad \alpha_{min} = \min \quad & \sum_{i=1}^{m}\sum_{i=1}^{m} x_{ij} \\
\text{s.t.} \quad & (5.2),\ (5.3),\ (5.4), \\
& \sum_{i=1}^{m}\sum_{j=1}^{m} (w_i \times cov_{ij}) x_{ij} \geq LB \\
& x_{ij} \in [0,1]\ \forall (i,j) \in I \times J,\ y_j \in [0,1]\ \forall\ j \in J,
\end{aligned}
$$

We observe that the first model is used to compute an upper bound on the maximum count of decision variables that can be incorporated into a feasible solution, with a value greater than or equal to the lower bound (LB). Moreover, the second model allows for the estimation of the minimum count of these variables that must be activated in the solution to attain an objective value greater than or equal to LB. These two models complement each other by providing complementary information about the subset of variables needed to achieve the desired objective value.

5.2.4 Searching for a better lower bound throughout the interval $\Big[\alpha_{min}, \alpha_{max}\Big]$

By adding constraints (5.11) and (5.12) to the relaxed model $\mathrm{MIP}_{\mathrm{FCMCLP}}$, with the contraint related to the best lower bound LB, we obtain

$$
\begin{aligned}
\mathrm{PR}_{\mathrm{FCMCLP}}^{\mathbf{LB}(k)}: \quad \max \quad & \sum_{i=1}^{m}\sum_{j=1}^{m} (w_i \times cov_{ij}) x_{ij} \\
\text{s.t.} \quad & (5.2),\ (5.3), (5.4), \\
& \alpha_{min} \leq \sum_{i=1}^{m}\sum_{j=1}^{m} x_{ij} \leq \alpha_{max} && (5.13) \\
& \sum_{i=1}^{m}\sum_{j=1}^{m} (w_i \times cov_{ij}) x_{ij} \geq LB && (5.14) \\
& x_{ij} \in [0,1], (i,j) \in I \times J, y_j \in \{0,1\},\ j \in J,
\end{aligned}
$$

An improved solution may be achieved by applying the following algorithm (Algo.8). At each step of the main loop, the algorithm uses the following model:

$$
\begin{aligned}
\mathrm{PR}_{\mathrm{FCMCLP}}^{LB(\delta=)}: \quad \max \quad & \sum_{i=1}^{m}\sum_{j=1}^{m} (w_i \times cov_{ij}) x_{ij} \\
\text{s.t.} \quad & (5.2),\ (5.3), (5.4), \\
& \sum_{i=1}^{m}\sum_{j=1}^{m} x_{ij} = \delta && (5.15) \\
& \sum_{i=1}^{m}\sum_{j=1}^{m} (w_i \times cov_{ij}) x_{ij} \geq LB && (5.16) \\
& x_{ij} \in [0,1], (i,j) \in I \times J, y_j \in \{0,1\},\ j \in J,
\end{aligned}
$$

where $\delta \in \Big[\lceil \alpha_{min} \rceil, \lfloor \alpha_{max} \rfloor\Big]$.

Input. An instance of P_{FCMCLP}.

Output. A (sub)optimal solution (X^0, Y^0) with its objective value LB.

Solve RP_{FCMCLP} with the truncated Cplex solver and let (X^0, Y^0) be its achieved optimal solution with objective value z^0.
Call the `Rounding Procedure` for providing a starting feasible solution of objective value LB.
Compute α_{max} (resp. α_{min}) by solving $\mathrm{P}_{\alpha_{max}}$ (resp. $\mathrm{P}_{\alpha_{min}}$).
Solve $\mathrm{PR}^{LB(\alpha)}_{\mathrm{FCMCLP}}$, call the `Rounding Procedure` on the provided solution. Let (X', Y') be the new rounded solution at hand with its objective function LB'.
if $(LB' \geq \mathrm{LB})$ **then**
    Update (X^0, Y^0) with $(X^{'}, Y^{'})$ and LB with LB'.
end if
Return (X^0, Y^0) with its objective value LB.

Algorithm 7: – A first solution in the interval $\left[\alpha_{min}, \alpha_{max}\right]$

Input. P_{FCMCLP}'s feasible solution (X^0, Y^0) with its objective value LB.

Output. An improved solution (X^0, Y^0) of value LB.

Compute α_{max} (resp. α_{min}) by solving $\mathrm{P}_{\alpha_{max}}$ (resp. $\mathrm{P}_{\alpha_{min}}$).
Set $\delta = \alpha_{min}$
repeat
    Solve $\mathrm{PR}^{LB(\delta=)}_{\mathrm{FCMCLP}}$, call the `Rounding Procedure` on the provided solution. Let (X', Y') be the new rounded solution at hand with its objective function LB'.
    if $(LB' \geq LB)$ **then**
        Update (X^0, Y^0) with $(X^{'}, Y^{'})$ and LB with LB'.
    end if
until $\delta = \alpha_{max}$
Return (X^0, Y^0) with its objective value LB.

Algorithm 8: – Enhancing LB on the interval $\left[\alpha_{min}, \alpha_{max}\right]$

5.2.5 Exploitation strategy

In this section, we employ an exploitation strategy, represented by an intensification operator, to refine the current solution through minor adjustments within the local neighborhood. One of the techniques utilized by exploitation strategies is the k-opt operator, which is a straightforward local search method that relies on fundamental, localized adjustments to the current solution. These adjustments can be understood as simple swaps within the y vector, representing items. The underlying assumption is that these swaps contribute to an improved solution quality. The series of swaps continues until a local optimum is reached, at which point the solution maintains the same objective value.

In the context of this study, a d-opt operation involves closing d presently open facilities j where $y_j = 1$, followed by the subsequent opening of k new facilities in a manner that sets their $y_t = 0$. It can be observed that when the value of d varies within a discrete interval, it becomes apparent that the d-opt operator can be extended to Variable Neighborhood Search (VNS). Such a procedure is known for its capability to escape local optima by systematically exploring diverse regions within the solution space. Instead of exclusively relying on a direct search associated with k-opt, one can use VNS to exploit its combined strengths of intensification (focused local search within specific neighborhoods) and diversification (exploration of various neighborhoods) to discover high-quality solutions for the problem at hand.

Input. S_0, a feasible solution of P_{FCMCLP}, and d_{max} (the maximum dropping parameter)
Output. An improved solution $S^\star$ for P_{FCMCLP}.

1: **repeat**
2:   Set $d \longleftarrow 1$ and $t \longleftarrow 0$
3:   **repeat**
4:     $S' \longleftarrow$ `Drop`(S_0, d) `// Swapping`
5:     $S'' \longleftarrow$ `LocalSearch`(S')
6:     **if** $\Big(z(S'') > z(S_0)\Big)$ **then**
7:       $S_0 \longleftarrow S''$ `// Make a swap`
8:       $d \longleftarrow 1$ `// Initial neighborhood`
9:     **else**
10:       $d \longleftarrow d + 1$ `// Next neighborhood`
11:     **end if**
12:   **until** $\Big(d = d_{max}\Big)$
13:   $t \longleftarrow$ `Time`$()$
14: **until** $\Big(t \geq T_{max}\Big)$
15: Return $S^\star \longleftarrow S_0$

Algorithm 9: – An exploitation search using VNS

Instead of directly using VNS, the step using `LocalSearch`(S') for a given solution (S') is replaced by Algorithm 8, with some arrangements. With this modification, Algorithm 9 resulting from it leads to the resolution of the problem P_{FCMCLP} by combining the enhanced basic rounding method with VNS. Indeed, given an initial solution S_0 (the current solution to improve), it starts by setting d to 1, representing the 1-opt or 1-Drop in step 10. In this step, a random solution S' is obtained by applying a first random drop or a series of drops for $d \geq 2$ using the procedure `Drop`(S_0, d). In step 11, a local search is performed on the provided drop solution; that is a partial solution for P_{FCMCLP}. Then, the local search procedure applied resembles a tabu search, where a local search involves replacing k opened facilities with other non-opened facilities such that the knapsack capacity related to the constraints (5.4) remains satisfied. Hence, a new solution, denoted S'', is provided and updates the next starting solution (step 13) whenever its objective value is better than that of the current starting solution, i.e.,

$z(S'') > z(S_0)$. If the aforementioned case holds, then the number of opened facilities is restarted by setting $k = 1$ (step 14). Otherwise, the next neighborhood is examined, where the number of opened facilities k is incremented according to the constraint related to the number of facilities to be opened in the problem (step 16).

Note that during each iteration of the local search, a tabu list is maintained by following these steps: (i) considering each set of neighboring solutions related to the current solution (which involves either dropping facilities from opened ones to closed ones or vice versa), (ii) evaluating the objective function for each neighboring solution, (iii) selecting the best neighboring solution that is not on the tabu list (or choosing the one with the least tabu tenure violation if necessary), (iv) updating the current solution to the selected neighbor, and (v) adding the move to the tabu list to prevent revisiting it in the near future.

Input. An instance of P_{FCMCLP}.
Output. A (sub)optimal solution $S^{\star}_{\text{FCMCLP}}$ for P_{FCMCLP}.

1: Call Algorithm 7 and let $S^{(0)}_{\text{FCMCLP}} = (X^0, Y^0)$ be the provided (initial) solution with its objective value LB_{FCMCLP}.
2: Call Algorithm 8 with $S^{(0)}_{\text{FCMCLP}}$ and let $S'_{\text{FCMCLP}} = (X'_{\text{FCMCLP}}, Y'_{\text{FCMCLP}})$ be the best solution reached with its objective value LB'_{FCMCLP}.
3: **if** $(LB'_{\text{FCMCLP}} > LB_{\text{FCMCLP}})$ **then**
4:     Set $S^{(0)}_{\text{FCMCLP}} = S'_{\text{FCMCLP}}$ and $LB_{\text{FCMCLP}} = LB'_{\text{FCMCLP}}$.
5: **end if**
6: Set $S^{*}_{\text{FCMCLP}} = S^{(0)}_{\text{FCMCLP}}$
7: **repeat**
8:     Set $d \longleftarrow 1$ and $t \longleftarrow 0$
9:     **repeat**
10:         $S' \longleftarrow$ `Drop`$(S^{(0)}_{\text{FCMCLP}}, d)$ `// Dropping and replacing`
11:         $S'' \longleftarrow$ `Algorithm` 8 with S' and its bound $z(S')$
12:         **if** $\left(z(S'') > z(S^{(0)}_{\text{FCMCLP}})\right)$ **then**
13:             $S^{(0)}_{\text{FCMCLP}} \longleftarrow S''$ `// Updating`
14:             $d \longleftarrow 1$ `// Initial neighborhood`
15:         **else**
16:             $d \longleftarrow d + 1$ `// Next neighborhood`
17:         **end if**
18:     **until** $\left(d = d_{max}\right)$
19:     Update S^{*}_{FCMCLP}, if necessary.
20:     $t \longleftarrow$ `Time()`
21: **until** $\left(t \geq T_{max}\right)$
22: Exit with the best solution S^{*}_{FCMCLP}.

Algorithm 10: – An Extended Rounding Strategy-Based Method: ERS-BM

5.2.6 An overview of the proposed method

Algorithm 10 describes the main steps of the Extended Rounding Strategy-Based Method (ERS-BM). The input of ERS-BM is an instance of P_{FCMCLP} and its output is a (near)optimal solution $S^{\star}_{\mathrm{FCMCLP}}$.

The algorithm starts by generating an initial solution (X^0, Y^0) with an objective value LB (line 1). The basic rounding procedure (Algorithm 7) is employed for reaching such a solution. Subsequently, a new solution $S'_{\mathrm{FCMCLP}} = (X'_{\mathrm{FCMCLP}}, Y'_{\mathrm{FCMCLP}})$ with an objective value LB' is provided by iteratively refining the search process within the constraint added; that is related to both cardinally constraints bounded by α_{min} and α_{max}, respectively. In this case, the initial solution with the current value LB is added as a new constraint to the current problem, i.e., $z(x) \geq LB$. Next, the incumbent solution $S^{(0)}_{\mathrm{FCMCLP}}$ is updated with S'_{FCMCLP} whenever the objective value $z(S'')$ becomes better than LB $(z(S^{(0)}_{\mathrm{FCMCLP}})$. This procedure relies on Algorithm 3, which simulates the VNS procedure, with a preference for a descent method to stop the search whenever the solution stagnates in local optima. Finally, the algorithm exits with the best solution S^{*}_{FCMCLP} reached so far.

5.3 Computational results

The purpose of this section is to assess the performance of the Extended Rounding Strategy-Based Algorithm (ERS-BA) by comparing its results with recent and competitive methods from the literature, as well as the state-of-the-art Cplex solver. A comprehensive evaluation is conducted using the same set of instances extracted from Atta et al. (2022). These sets contains various values of k, representing the number of facilities to be opened, and the groups are categorized based on the assigned values of m. Additional information regarding the parameter values can be found in Table 3.1. The objective of the study is to assess how well ERS-BA performs in comparison to existing methods of the literature and the Cplex solver.

The parameters reported in Table 3.1 can be explained as follows. The first column contains values related to m, representing the number of customers to be served. Specifically, m takes on discrete values within the range $\{324; 402; 500; 700; 708; 818; 900\}$. The second column corresponds to the number of open facilities, denoted as k. The values of k depend on the radius of the coverage area of circles associated with each facility. In this case, R represents the radius, $R + f_r$ indicates the fuzzy radius, and f_d represents the additional fuzzy value applied to the distance between each customer and the installed facility. Finally, the values in the last column indicate the supply capacity of each open facility within the network, which can be either 4500 or 9500.

5.3.1 The quality of the starting rounding bound: BRP

It is common for a sequential method to focus on how to generate at least moderately good initial solutions because such an approach can sometimes play a crucial role in its convergence. Of course, both exploitation and exploration operators can also be employed as strategies to either enhance the quality of the solution at hand or diversify the search process into un-visited regions, thus preventing premature convergence.

In this section, the quality of the lower bounds obtained by the BRP (Basic Rounding Procedure, Section 5.2.2) algorithm is analyzed. As highlighted in Table 5.1, two groups of instances are considered: (i) instances qualified as medium to large-sized, and (ii) very large-scale instances, with $k \geq 4$. For the initial set of instances, we conducted the BRP method. Nevertheless, considering the substantial execution time demanded by this method when applied to instances in the second group, we made the following modifications:

1. Set $P_{\text{reduce}} = \text{MIP}_{\text{FCMCLP}}$.
2. Solve P_{reduce} with the Cplex solver to obtain the solution $(\underline{x}, \underline{y}, \underline{z})$.
3. Set the value of each $\underline{y}_j$ to 1 in $\underline{y}$ if $\underline{y}_j = 1$.
4. Repeat
   (a) Choose the variable $\underline{x}it$ with the highest fractional value among the fractional variables, and label it as $\underline{x}i_0t$.
   (b) Select the best facility j_0 for customer i_0 based on:
      i. Verify if the remaining supply is non-negative: $S^{res}{}_{j_0} \geq 0$, i.e., $S^{res}{}_{j_0} = S_{j_0} - w(i_0) * cov(i_0, j_0)$.
      ii. The selected facility j_0 fulfills the following condition: $j_0 = \arg \max_{1 \leq j \leq n} \left\{ w_{(i_0)} * cov(i_0, j) \right\}$.
      iii. If both conditions hold, set $\underline{x}_{i_0 j} = 0$ for all $j \in J, j \neq j_0$. Otherwise, set $\underline{x}_{i_0 t} = 0$.
5. Until all decision variables are within the range $x_{ij} \in \{0, 1\}$.

Table 5.1: Solution quality of BRPs' starting upper bounds on all instances

#Inst.		UB_{Cplex}	LB_{Cplex}	LB_{BRP}	A(I)
324	$1 \leq k \leq 7$	85.22	82.71	81.95	0.963022
402	$1 \leq k \leq 9$	84.61	83.14	82.61	0.979970
500	$1 \leq k \leq 13$	86.48	83.17	82.95	0.963529
	Av.	85.44	83.14	82.50	0.968840
708	$1 \leq k \leq 15$	84.99	na	79.21	na
818	$1 \leq k \leq 15$	80.80	na	76.39	na
700	$1 \leq k \leq 12$	79.58	73.02	73.43	0.938422
900	$1 \leq k \leq 13$	74.53	73.56	73.85	0.994567
	Av. Solved	79.975	73.29	75.72	0.966494
	Av. All	82.7075	78.215	79.11	0.967667

Table 5.1 reports the results obtained by BRP with its improvement in the cardinality constraint interval $[\alpha_{\min}, \alpha_{\max}]$ for all groups of instances. The table organizes instances into two sets, represented by two blocks: the first set (block) contains the three first group of instances and the second one contains the rest of the groups. In the table, columns 1 and 2 display the instance details, column 3 (resp. column 4) shows the upper (resp. lower) bound related to the Cplex solver (noted UB_{Cplex}, resp. LB_{Cplex}) with a fixed runtime limit, column 5 tallies the solution value obtained by BRP (noted LB_{BRP}), and the last column displays the experimental approximation ratio of an instance I (noted A(I)) between the lower bound achieved by BRP and Cplex's upper bound, calculated as follows: A(I)$= \frac{\text{LB}_{\text{BRP}}}{\text{UB}_{\text{Cplex}}}$. As mentioned earlier, BRP involves solving a series of MIPs, resulting in varying runtimes ranging from one second to five minutes for the tested instances. In the following, we provide an analysis of the results presented in Table5.1:

- For the first set (block) of instances, columns 4 and 5 indicate that BRP is able to provide an average approximation ratio varying from 81.95 (Group 324,402,500) to 82.95 (Group 708,818,700,900). Line 4 confirms the good average approximation ratio since the global value is equal to 82.50, representing an average percentage relative value of 0.968840.

When comparing the results to those achieved by the Cplex solver, one can observe that despite the simplicity of such an approach, it is able to compete to achieve high-quality solutions. Indeed, as observed from line 4 (the first block), the Cplex solver achieves an average lower bound of 83.14 while BRP is able to provide a near-average bound of 82.50.

- In the case related to the second set of instances (second block), the average quality of the objective functions (lower bounds) appears even more intriguing. Indeed, we observe a variation in the experimental approximation ranging between 79.21 and 73.85. we can also note a clear improvement in the overall enhancement of the lower bounds for BRP when these values are compared to those provided by Cplex.
- Finally, it is worth highlighting the average global experimental approximation ratio achieved by BRP (0.967667: last column, last line), which remains interesting, underlining its effectiveness.

The above observations suggest the potential benefit of combining such a procedure with other exploitation and exploration strategies. However, to demonstrate the value of using such a procedure in an extensive search, Figure 5.1 illustrates the variation in the percentage positive ratio, $\frac{UB(I)}{LB(I)}$, across all tested instances, in comparison to the global average curve.

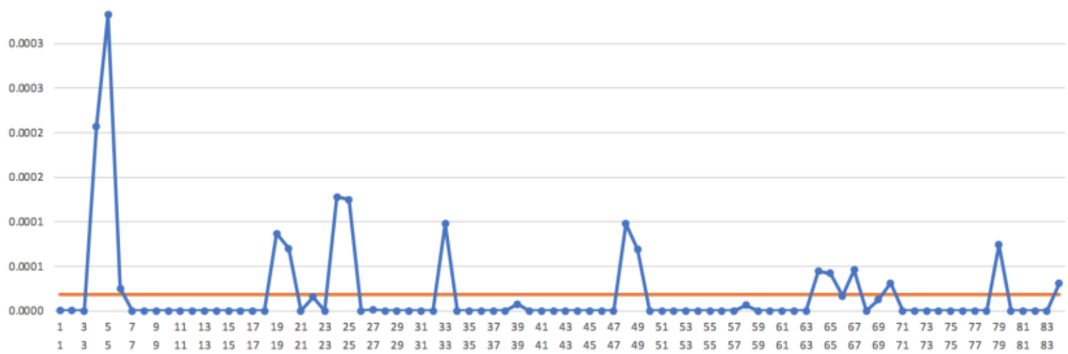

Figure 5.1: Variation of the relative approximation ratio in percentage according to overall tested instances.

5.3.1.1 Effect of the parameters used by VNS

The performance of VNS frequently depends on the selection and configuration of its parameters. In this section, the impact of these parameters is explored, with a particular focus on the influence of incorporating a variation in the k-opt operator.

As a descent method with the variation of d is applied, the results for a subset of instances from the overall groups considered in the study are examined. Table 5.2 reports all the results obtained by varying the parameter d from 1 to 4, based on the number of facilities to be opened for each instance. Of course, various values for d were tested, and only those demonstrating effectiveness in fine-tuning VNS are retained.

Table 5.2 reports ERS-BAs' results, where columns 1 and 2 display the instance information, column 3 displays the BRP's starting lower bound (percentage coverage), columns 4 and 5 tally the bounds related to both cardinality constraints, and the rest of the columns show the best average lower bounds achieved by ERS-BA according to overall values related to the size of the te number of dropping depots. Finally, the last line of the table tallies the global average lower bound of all tested instances for each version of ERS-BA. In what follows, we comment on the results of Table 5.2:

Table 5.2: Solution quality of BRPs' starting lower bounds on all instances

Instance			Cardinality constraints		Variation of the parameter d			
m	k	LB_{BRP}	α_{min}	α_{max}	1-opt	2-opt	3-opt	4-opt
324	4	90.32	190.40	324.00	93.25	–	–	–
	5	95.45	227.00	324.00	98.72	–	–	–
	6	99.48	242.36	324.00	99.75	99.75	99.77	–
500	3	65.86	97.37	500.00	65.86	66.44	–	–
	4	79.18	165.30	500.00	79.18	79.59	–	–
	5	86.71	245.10	500.00	86.71	87.17	–	–
	6	90.38	279.78	500.00	91.78	92.17	–	–
	10	99.19	403.80	500.00	99.49	–	–	–
700	6	86.66	571.68	700.00	86.66	–	–	–
	7	87.21	576.36	700.00	87.21	87.81	89.64	–
708	4	69.93	191.20	708.00	69.93	70.11	70.46	–
	5	99.23	575.22	708.00	99.23	99.23	99.54	–
	Av.	87.47	313.80	524.00	88.15	88.37	88.57	88.57

- For all discrete values of d ranging from 1 to 4, each version of ERS-BA consistently improves the average bounds based on the sizes of the instances within their respective groups. For all discrete values of d ranging from 1 to 4, each version of ERS-BA consistently enhances the average bounds based on the sizes of the instances within their respective groups.

- As observed in the table, the higher the value of d, the more apparent the improvement in bounds becomes. In this case, the enhancement varies from 88.15 with $d = 1$ (contrasted with the average value of BRP at 87.47) to 88.57 with $d = 3$ (compared to the average BRP value of 87.47).

- For all instances, ERS-BA with $d = 4$ (except for instance 500.3, which contains 3 open facilities) shows superior performance when comparing its values to those achieved by other versions of the algorithm with $d = 3,\ 2$, and 1, in decreasing order. Specifically, it attains an average value of 88.57, as highlighted above.

- ERS-BA with $d = 3$ shows favorable performance, as observed across all considered instances. Indeed, for $d = 4$, all provided lower bounds match those achieved by ERS-BA with $d = 3$. Therefore, it is crucial to optimize the average runtime without compromising the quality of the obtained solutions

Note that, as we seek the version able of achieving high-quality solutions (maximizing objective values) without significantly increasing the average runtime, we then choose to set $d = 3$ (the maximum value in this work) for the rest of the experimental part.

5.3.1.2 ERS-BA versus other methods of the literature

As underlined at the staring of this section, in order to perform a comparison with other methods of the literature, we grouped the three first groups of instances into one set; that is considered as a set containing moderate-sized instances. The second set contains the rest of the instances that includes last four groups, whee these instances are considered as large-sized instances.

All results related to the average bounds of each group are reported in Table 5.3. Each line of the table displays these average values while the last line of each set (block) reports the global average values achieved by the proposed ERS-BA and those achieved by two more recent algorithms in the literature (Aider et al. (2023); Atta et al. (2022)) (denoted Best_{Lit}).

Table 5.3: Performance of ERS-BA versus both Cplex and the best method of the literature (Aider et al., 2023): medium-sized instances

Instance				This work	
m	k	$\mathrm{LB}_{\mathrm{Cplex}}$	$\mathrm{LB}_{\mathrm{GWO}}$	$\mathrm{LB}_{\mathrm{BRP}}$	ERS-BA
324	1	37.03	37.03	37.03	37.03
	2	65.14	65.14	65.14	65.14
	3	86.24	86.24	86.24	86.24
	4	92.58	93.1	90.32	93.25
	5	98.24	99.1	95.45	99.10
	6	99.72	99.8411	99.48	99.84
	7	99.99	99.96	99.99	99.99
	Av.	82.71	82.92	81.95	82.942814
402	1	28.15	28.1516	28.15	28.15
	2	56.1	56.10	56.10	56.10
	3	77.25	77.25	77.25	77.25
	4	91.59	91.8112	90.72	91.75
	5	96.74	96.9803	95.45	96.85
	6	99.39	99.1192	97.36	99.29
	7	99.47	99.7136	98.45	99.94
	8	99.86	99.7377	99.99	100.00
	9	99.71	99.7716	100.00	100.00
	Av.	83.14	83.18	82.61	83.260877
500	1	22.83	22.834	22.84	22.84
	2	45.67	45.66	45.67	45.67
	3	66.44	66.44	65.86	66.44
	4	80.01	80.19	79.18	79.77
	5	87.11	86.7523	86.71	88.30
	6	91.68	93.0397	90.38	92.61
	7	94.9	96.06	93.83	95.49
	8	97	98.5781	96.17	98.86
	9	97.54	99.0321	98.79	99.05
	10	99.42	99.712	99.19	99.49
	11	98.69	99.5544	99.79	99.94
	12	99.97	99.61	99.99	99.99
	13	99.98	100	99.99	99.99
	Av.	83.17	83.65	82.95	83.73
	Av. Global	83.14	83.43	82.78	83.51

Tables 5.3 and 5.4 report the solution values achieved by all compared methods: the Cplex solver, the method designed in Aider et al. (2023) (HP-BA: Hybrid Population-Based Algorithm), and the proposed algorithm ERS-BA on all instances. In each table, columns 1 and 2 of the table shows the instance's label, column 3 reports the Cplex solver's integer bound ($\mathrm{LB}_{\mathrm{Cplex}}$), column 4 tallies the best lower bound reached by HP-BA ($\mathrm{LB}_{\mathrm{HP-BA}}$) extracted from Aider et al. (2023), column 4 tallies BRP's lower bound ($\mathrm{LB}_{\mathrm{BRP}}$), and column 5 shows the best bounds achieved by the proposed method ERS-BA (LB_{ERS-BA}). From Tables 5.3 and 5.4, we can observe what follows:

1) Medium-sized instances (Table 5.3):

- Cplex versus other solution procedures: Cplex dominates BRP, as already discussed, but the gap between both methods remains small. However, Cplex is less efficient in searching for high-quality solutions since HP-BA establishes a higher (better) global average lower bound: 83.43 compared to 83.14.
- ERS-BA versus other methods: It is evident that ERS-BA outperforms both Cplex and HP-BA. For this group of instances, it achieves new (lower) bounds, as observed from the last column of Table 5.3.
- The global average achieved by ERS-BA, as shown in the last line of Table 5.3, is equal to 83.51, is greater than both Cplex average bound and HP-BA average bound.

2) Large-scale instances (Table 5.4):

- Cplex versus other solution procedures: Cplex maintains superiority in generating lower

Table 5.4: Performance of ERS-BA versus both Cplex and the best method of the literature (Aider et al., 2023): large-scale instances

Instance				This work	
m	k	LB_{Cplex}	LB_{GWO}	LB_{BRP}	ERS-BA
708	1	18.6	18.604	18.60	18.60
	2	37.2	37.1861	37.20	37.20
	3	55.8	55.6383	55.80	55.80
	4	70.86	70.61	69.93	70.46
	5	80.89	81.4843	78.81	80.73
	6	85.28	86.51	82.43	85.77
	7	na	89.1147	83.54	88.34
	8	na	91.82	87.43	92.02
	9	na	94.62	90.91	94.00
	10	na	97.66	93.04	95.75
	11	na	98.39	96.23	98.27
	12	na	98.391	96.62	98.38
	13	na	99.441	98.82	99.28
	14	na	99.51	99.23	99.54
	15	na	99.29	99.55	99.73
	Av.	na	81.22	79.21	80.92
818	1	15.43	15.43	15.43	15.43
	2	30.86	30.85	30.86	30.86
	3	46.28	46.1	46.26	46.26
	4	59.97	59.74	58.20	59.69
	5	72.42	72	71.75	72.27
	6	81.04	81	80.57	81.00
	7	84.3	85.61	83.55	85.02
	8	na	89.35	89.35	89.35
	9	na	92.26	92.26	92.26
	10	na	93.84	93.84	93.84
	11	na	95.46	95.46	95.46
	12	na	96.39	96.40	96.40
	13	na	97.01	97.01	97.01
	14	na	98.82	97.07	98.11
	15	na	99.57	97.85	99.13
	Av.	na	76.90	76.39	76.81
700	1	17.01	17.011	17.01	17.01
	2	34.02	34.023	34.02	34.02
	3	51.03	51.03	51.03	51.03
	4	68.04	68.046	68.046	68.046
	5	82.38	83.8	84.75	84.75
	6	85.88	89.09	86.66	90.39
	7	87.06	91.32	87.21	91.32
	8	88.64	92.73	89.05	92.50
	9	89.3	93.55	92.13	93.78
	10	90.23	94.28	89.34	94.30
	11	91.03	94.61	90.71	94.95
	12	91.67	95.51	91.18	95.44
	Av.	73.02	75.42	73.43	75.63
900	1	13.18	13.18	13.18	13.18
	2	26.35	26.35	26.35	26.35
	3	39.53	39.53	39.53	39.53
	4	52.7	52.7	52.70	52.70
	5	65.88	65.86	65.87	65.87
	6	79.05	79.04	79.03	79.04
	7	92.22	92.16	92.23	92.23
	8	97.18	98.3	96.74	97.18
	9	97.43	98.431	98.43	98.43
	10	97.77	98.943	98.94	98.94
	11	98.21	99.261	99.26	99.26
	12	98.28	99.52	99.52	99.52
	13	98.53	99.33	98.28	98.28
	Av.	73.56	74.04	73.85	73.89
	AV. Global	na	76.89	75.72	76.81

bounds when compared to BRP, which is expected for an exact algorithm. However, the two other methods, HP-BA and ERS-BA, remain better than Cplex, as their overall averages bounds are greater than that reached by Cplex. However, we note that Cplex is able to reach better bounds than those provided by ERS-BA, as observed in Table 5.4.

- Behavior of ERS-BA versus other methods: We observe that while ERS-BA achieves several new lower bounds compared to those provided by HP-BA, it remains the case that only the group with 700 nodes witnesses the proposed method dominating over HP-BA. However, when focusing on the overall average related to lower bounds for this set of instances, we observe that the values remain close: 76.89 for HP-BA compared to 76.81 for ERS-BA

5.4 Conclusion

In this chapter, we addressed the fuzzy capacitated maximal covering location problem, which involves the optimal selection of facilities to serve customers in a network. Such a problem was tackled by an algorithm that is based upon an extended rounding strategy. This algorithm uses a specialized basic rounding procedure to establish an initial feasible solution. Subsequently, an improved bound is provided by introducing constraints related to the cardinally constraints associated with the decision variables used, reflecting the nodes to open. By using these cardinally constraints, a variable neighborhood descent search was proposed to enhance the quality of the solutions at hand. Finally, to evaluate the effectiveness of the proposed method, the results were compared with those achieved by recent methods in the literature and the state-of-the-art Cplex solver. Experiments were conducted on a set of benchmark instances commonly used in the literature.A comparative analysis that demonstrated promising and encouraging results will be provided later.

General Conclusion and Perspectives

In this thesis, a deep resolution study was included to solve one of the NP-Hard variant of the well-known binary optimization problem, the maximal covering location problem, called the Fuzzy Capacitated Maximal Covering Location Problem, namely (FCMCLP).
The FCMCLP involves determining the optimal selection of facilities to serve customers in a network. The problem considers fuzzy parameters such as the coverage degree of facilities and the distance between customers, adding an element of uncertainty to the decision-making process.
In order to solve the fuzzy capacitated maximal covering location problem, we proposed three different approaches based upon hybrid metaheuristics and evolutionary algorithms.

The 1^{st} chapter, shows general issues and definitions that concerns the combinatorial optimization and the different resolution techniques proposed and used in the literature. In the 2^{nd} chapter, we presented the Maximal Covering Location problem and its well-known variants, and showed their different mathematical formulation.
In the 3^{rd} chapter, Two versions of a hybrid population-based algorithm are designed for tackling the aforementioned problem: the first version, a grey wolf optimizer strategy combined with exploration and exploitation strategies, and in the second one, a modified grey wolf optimizer combined with a series of local searches and an upper bound delimiting wolf movement. Each version of the method enhances the quality of the solutions belonging to the current population by adding a series of local searches. Next, we proposed to add the Drop and Rebuild Operator to provide a manner able to drive the search process through other un-visited sub-spaces. Finally, we tested the performance of the hybrid population-based algorithm on a set of benchmark instances, where its achieved results were compared to those provided by the state-of-the-art Cplex solver and to the best results extracted from a recent paper in the literature. Many new improved results have been discovered.

The 4^{th} chapter was a basic study of the effect of the discrete grey wolf optimizer for solving the fuzzy capacitated maximal covering location problem. First, an initial population was constructed by tailoring a random greedy procedure. Second, as an intensification strategy, we added a 2-opt operator, which is even based upon basic local modifications of the solution at hand. Often, the operator sequentially makes a series of swaps as long as the quality of the induced solution is enhanced.Last, aiming to diversify the search space, the drop and rebuild operator was incorporated. Finally, the evolutionary algorithm was evaluated on the same set of benchmark instances, where encouraging results have been provided.

The last chapter (refer to chapter 5), we designed an extended rounding strategy-based algorithm to solve the FCMCLP. The algorithm starts by applying the basic rounding procedure, which establishes an initial feasible solution for the problem. Subsequently, an interval search is conducted, incorporating both lower and upper limits, to constrain the search process and improve the quality of the solution. At each iteration, in order to refine the solution, the algorithm follows an iterative search approach, resembling a descent method. The effectiveness of the proposed method was shown through its provided results, by comparing them to those achieved by recent methods in the literature as well as the state-of-the-art Cplex solver.

Through this study on the localization problem, we believe that there are several opportunities for further research involving effective methods capable of improving the proposed method. First, we believe that some pre-processing procedures can be tailored for the studied problem, where tights bounds and valid constraints can be added for reducing the size of the initial instance. In this case, the population-based method can be augmented with adding a fix and solve procedure. Second, the Benders' decomposition can be applied to the studied problem, where the two types of variables can cooperate to provide an iterative method based on the addition of a series of Benders constraints, which can be combined with other valid constraints. Finally, a parallel approach can be considered as another option to solve this problem; in this case, one can imagine partitioning the neighborhoods to design a cooperative parallel method, which can improve the quality of the solutions while reducing the global run-time.

List of scientific publications related to the thesis

Journal articles published

Méziane Aider, Imene Dey, Mhand Hifi. "A hybrid population-based algorithm for solving the fuzzy capacitated maximal covering location problem". Computers & Industrial Engineering(2023),Volume 177,108982, doi: https://doi.org/10.1016/j.cie.2023.108982

Conference paper published

M. Aïder, I. Dey and M. Hifi, "A Population-Based Algorithm for Solving the Fuzzy Capacitated Maximal Covering Location Problem," 2022 9th International Conference on Soft Computing & Machine Intelligence (ISCMI), Toronto, ON, Canada, 2022, pp. 91-96, doi: 10.1109/ISCMI56532.2022.10068459.

Communications

Méziane Aider, Imene Dey, Mhand Hifi, "Contribution à l'optimisation dans les réseaux", Journée EPROAD, Laboratoire ROD, UR 4669, Salle BC02 – Pôle Sciences,33 rue Saint Leu, Amiens, 7 juillet 2022.

Méziane Aider, Imene Dey, Mhand Hifi, "A hybrid population-based algorithm for solving the fuzzy capacitated maximal covering location problem", Journée JDROM, Laboratoire LAROMAD, Faculté de Mathématiques, USTHB, 03-04 Mai 2023.

Méziane Aider, Imene Dey, Mhand Hifi, "The two-edge disjoint survivable network design problem with relays", Journée JDROM, Laboratoire LAROMAD, Faculté de Mathématiques, USTHB, 16 Décembre 2021.

Bibliography

Aider, M., Dey, I., and Hifi, M. (2023). A hybrid population-based algorithm for solving the fuzzy capacitated maximal covering location problem. *Computers Industrial Engineering*, 177:108982.

Atta, S., Mahapatra, P., and Mukhopadhyay, A. (2018). Solving maximal covering location problem using genetic algorithm with local refinement. *Soft Computing*, 22:3891–3906.

Atta, S., Ranjan Priya Sinh, M., and Mukhopadhyay, A. (2022). Solving a new variant of the capacitated maximal covering location problem with fuzzy coverage area using metaheuristic approaches. *Computers & Industrial Engineering. To appear*, 170:108315.

Aïder, M., Dey, I., and Hifi, M. (2022). A population-based algorithm for solving the fuzzy capacitated maximal covering location problem. In *2022 9th International Conference on Soft Computing Machine Intelligence (ISCMI)*, pages 91–96.

Church, R. and ReVelle, C. (1974). The maximal covering location problem. In *Papers of the Regional Science Association*, pages 110–118. Springer-Verlag.

Clerc, M. (2010). *Particle swarm optimization*, volume 93. John Wiley & Sons.

Cook, W. J., Applegate, D. L., Bixby, R. E., and Chvatal, V. (2011). *The traveling salesman problem: a computational study.* Princeton university press.

Davari, S., Zarandi, M., and Hemmati, A. (2011). Maximal covering location problem (mclp) with fuzzy travel times. *Expert Systems with Applications*, 38:14535–14541.

Davari, S., Zarandi, M., and Turksen, I. (2013). A greedy variable neighborhood search heuristic for the maximal covering location problem with fuzzy coverage radii. *Knowledge-Based Systems*, 41:68–76.

Drakulic, D., Takaci, A., and Maric, M. (2016). New model of maximal covering location problem with fuzzy conditions. *Computing and Informatics*, 35:635–652.

Drezner, Z. and Hamacher, H. W. (2002). *Facility Location Applications and Theory.* Springer Berlin, Heidelberg.

ElKady, S. K. and Abdelsalam, H. M. (2016). A modified particle swarm optimization algorithm for solving capacitated maximal covering location problem in healthcare systems. *Applications of Intelligent Optimization in Biology and Medicine*, pages 117–133.

Farahani, R., Asgari, N., Heidari, N., Hosseininia, M., and Goh, M. (2012). Covering problems in facility location: A review. *Computers & Industrial Engineering*, 62:114–123.

Flood, M. M. (1956). The traveling-salesman problem. *Operations Research*, 4(1):61–75.

Frank, P. and Lieselot, V. (2007). Aggregation without loss of optimality in competitive location models. *Networks and Spatial Economics*, 7:3–18.

Galvao, R. and ReVelle, C. (1996). A lagrangean heuristic for the maximal covering location problem. *European Journal of Operational Research*, 88:114–123.

Gendreau, M., Potvin, J.-Y., et al. (2010). *Handbook of metaheuristics*, volume 2. Springer.

Glover, F. (1989). Tabu search-part i. *ORSA Journal on Computing*, 1(3):190–206.

Hifi, M. (2002). An iterative rounding search-based algorithm for the disjunctively constrained knapsack problem. *Engineering Optimization*, 46(8):1109–1122.

Lagos, C., Guerrero, G., Cabrera, E., Niklander, S., Johnson, F., Paredes, F., and Vega, J. (2016). A matheuristic approach combining local search and mathematical programming. *Scientific Programming*, 2016:1–1.

Laguna, M., Barnes, J. W., and Glover, F. W. (1991). Tabu search methods for a single machine scheduling problem. *Journal of Intelligent Manufacturing*, 2:63–73.

Land, A. H. and Doig, A. G. (2010). *An automatic method for solving discrete programming problems*. Springer.

Lawler, E. L., Lenstra, J. K., Kan, A. R., and Shmoys, D. B. (1986). The traveling salesman problem: a guided tour of combinatorial optimization. *The Journal of the Operational Research Society*, 37(5):535.

Lee, G. and Murray, A. T. (2010). Maximal covering with network survivability requirements in wireless mesh networks. *Computers, Environment and Urban Systems*, 34:49–57.

Levitin, A. (2005). The design and analysis of algorithms.

Li, X., Zhao, Z., Zhu, X., and Wyatt, T. (2011). Covering models and optimization techniques for emergency response facility location and planning: a review. *Mathematical Methods of Operations Research*, 74:281–310.

Little, J. D., Murty, K. G., Sweeney, D. W., and Karel, C. (1963). An algorithm for the traveling salesman problem. *Operations research*, 11(6):972–989.

Lorena, L. and Pereira, M. (2002). A lagrangean/surrogate heuristic for the maximal covering location problem using hilman's edition. *International Journal of Industrial Engineering*, 9:57–67.

Maniezzo, V., Stützle, T., and Voß, S. (2021). *Matheuristics*. Springer.

Muro, C., Escobedo, R., Spector, L., and Coppinger, R. (2011). New model of maximal covering location problem with fuzzy conditions. *Behavioural processes*, 88:192–197.

Murray, A. T., Kim, K., Davis, J. W., Machiraju, R., and Parent, R. (2007). Coverage optimization to support security monitoring. *Computers, Environment and Urban Systems*, 31:133–147.

Nodarse, C. P., Fajardo, J., and Rosete-Suàrez, A. (2019). Multi-coverage dynamic maximal covering location problem. *Investigacion Operacional*, 40(1):140–149.

Owen, S. H. and Daskin, M. S. (1998). Strategic facility location: A review. *European Journal of Operational Research*, 111(3):423–447.

Papadimitriou, C. H. and Steiglitz, K. (1998). *Combinatorial optimization: algorithms and complexity*. Courier Corporation.

Pirkul, H. and Schilling, D. (1991). The maximal covering location problem with capacities on total workload. *Management Science*, 37:233–248.

Pulver, A., Wei, R., and Mann, C. (2016). Locating aed enabled medical drones to enhance cardiac arrest response times. *Prehospital Emergency Care*, 20(3):378–389.

Reihaneh, M. and Ghoniem, A. (2018). A branch-cut-and-price algorithm for the generalized vehicle routing problem. *Journal of the Operational Research Society*, 69(2):307–318.

ReVelle, C., Scholssberg, M., and Williams, J. (2008). Solving the maximal covering location problem with heuristic concentration. *Computers & Operations Research*, 35(2):427–435.

Seyedali Mirjalili, S. M. M. and Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61.

Snyder, S. and Haight, R. (2016). Application of the maximal covering location problem to habitat reserve site selection: A review. *International Regional Science Review*, 39:28–4.

Takaci, A., Maric, M., and Drakulic, D. (2012). The role of fuzzy sets in improving maximal covering location problem. pages 103–106.

Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*. John Wiley & Sons.

Tanksale, A. and Jha, J. (2018). Benders decomposition approach with heuristic improvements for the robust foodgrain supply network design problem. *Journal of the Operational Research Society*, 71(1):16–36.

Voss, S., Stutzle, T., and Maniezzo, V. (2009). *MATHEURISTICS: Hybridizing metaheuristics and mathematical programming*. Springer.

Wang, F. (2020). A successive domain-reduction scheme for linearly constrained quadratic integer programming problems. *Journal of the Operational Research Society*, pages 1–14.

Williamson, D. P. and Shmoys, D. B. (2011). *The design of approximation algorithms*. Cambridge university press.

Xue, Y., Wen, Z., Ji, X., Bressers, H. T. A., and Zhang, C. (2017). Location optimization of urban mining facilities with maximal covering model in gis: A case of china. *Industrial Ecology*, 21(4):913–923.

Zarandi, M., Davari, S., and Sisakht, S. (2011). The large scale maximal covering location problem. *Scientia Iranica*, 18:1564–1570.